# MODERN STATISTICAL METHODS FOR OPTIMIZATION AND CHANGE-POINT DETECTION

A Dissertation
Presented to
The Academic Faculty

By

Zhehui Chen

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and System Engineering

Georgia Institute of Technology

August 2021

**MODERN STATISTICAL METHODS FOR OPTIMIZATION AND CHANGE-POINT DETECTION**

Approved by:


Dr. C.F. Jeff Wu
School of Industrial and System Engineering
*Georgia Institute of Technology*

Dr. Tuo Zhao
School of Industrial and System Engineering
*Georgia Institute of Technology*

Dr. Roshan V. Joseph
School of Industrial and System Engineering
*Georgia Institute of Technology*

Dr. Yao Xie Chen
School of Industrial and System Engineering
*Georgia Institute of Technology*

Dr. Simon Tsz Fung Mak
Department of Statistical Science
*Duke University*


Date Approved: April 29, 2021

Stay Hungry Stay Foolish

*Steve Jobs*

To my parents, Linjie Chen and Linyan Ruan, for their love, encouragement and support.

**ACKNOWLEDGEMENTS**

I would like to express my deepest appreciation to my advisors Professor C. F. Jeff Wu and Professor Tuo Zhao at Georgia Tech for their patience, motivation and passion in research, and valuable advices in life. Professor Wu's mentorship is not only broadening my perspectives on research, but also keeps nurturing me and developing me as a critical thinker; Professor Zhao guides me to work on the cutting-edge machine learning research and helps me develop necessary skills. I am deeply grateful for their continuous support and mentorships. This work would not have been possible without their help.

I am also grateful to the members of my committee - Professor Yao Xie, Professor Roshan Joseph, and Professor Simon Mak - for their support throughout my doctoral program, and constructive comments and suggestions for my doctoral dissertation work. Special thanks go to Professor Simon Mak for supervising me since he started the post-doc position at Georgia Tech.

My special thank to the wonderful faculties, staffs, and students at Georgia Tech. I would like to thank all my past and present lab mates: Dr. Yang Lin, Dr. Xingguo Li, Dr. Chih-Li Sung, Dr. Wenjia Wang, Dr. Yuanshuo Zhao, Dr. Li-Hsiang Lin, Dr. Liang Ding, Tianyi Liu, Haoming Jiang, Minshuo Chen, Yujia Xie, Yuyang Shi, Yan Li, Chen Liang, Henry Yuchi and Shangkun Wang for enlightening conversations and research discussions. I would also like to thank my friends that been with me at Georgia Tech: Jialei Chen, Liyan Xie, Guanyi Wang, Wanrong Zhang, Yu Yang, Zheng Wang, Weiyang Liu, and Hanbin Ying. Each of you helped me learn and grow in the past five years.

Last but definitely not least, my heartfelt thanks go to my parents. Without your love, encouragement, and most of all patience, I would not be able to achieve this important milestone in my life. This thesis is dedicated to them.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# SUMMARY

Optimization and change-point detection are two important problems in modern science and engineering. With remarkable advancements in computer engineering and electrical engineering, one of the challenging parts in these two problems is how to deal with big and high-dimensional data, *e.g.*, streaming data. The main focus of this thesis is to use recent statistical tools to study and develop efficient optimization algorithms for big data under different settings, and design effective and scalable change-point detection frameworks for network data.

Chapter 1 of the thesis studies the partial least squares (PLS) with streaming data, which can be efficiently solved by a stochastic generalized Hebbian algorithm (SGHA). Theoretically, we characterize the three phases of the SGHA by diffusion processes, and establish the corresponding global rates of convergence to the global optima. Empirically, we conduct some numerical experiments and the results also support our theory. In Chapter 2, we then study the generalized eigenvalue (GEV) decomposition problem, a general form of PLS. We first show that the Lagrangian function of GEV enjoys two properties: 1.Equilibria are either stable or unstable; 2.Stable equilibria correspond to the global optima of the original GEV problem. Inspired by these nice properties, we design a simple, efficient, and stochastic primal-dual algorithm solving the online GEV problem. By diffusion approximations, we obtain the first sample complexity result for the online GEV problem. Numerical results are also provided to support our theory.

The goal of Chapter 3 is how to improve a sequential design strategy for the global optimization of black-box functions, called expected improvement (EI). We first identify the over-greediness issue of EI. To address this problem, we propose a new hierarchical expectation improvement (HEI) framework. HEI preserves a closed-form acquisition function, and encourages exploration of the optimization space. We then introduce hyperparameter estimation methods which allow HEI to mimic a fully Bayesian optimization procedure,

while avoiding expensive Markov-chain Monte Carlo sampling steps. We prove the global convergence of HEI over a broad function space, and establish near-minimax convergence rates under certain prior specifications. Numerical experiments show the improvement of HEI over existing Bayesian optimization methods, for synthetic functions and a semiconductor manufacturing optimization problem.

Chapter 4 then studies a bilevel optimization problem, which contains a follower problem and a leader problem. Taking adversarial training as an example, we propose a generic learning-to-learn (L2L) method to solve it. The key idea of L2L is that instead of applying hand-designed algorithms, *e.g.*, stochastic gradient methods, to the follower problem, we learn an optimizer parametrized by a neural network. Meanwhile, the leader learns a robust model to defend the malicious adversarial attacks generated by the learned optimizer. Our experiments over CIFAR datasets demonstrate that L2L improves upon existing methods in both robust accuracy and computational efficiency. Moreover, we show that the proposed L2L method also works for other bilevel problems in machine learning such as adversarial interpolation training and general adversarial imitation learning.

Chapter 5 considers a change-point detection problem with network data, and designs a new Conditional AutoRegressive Detection (CARD) monitoring system, which models spatial correlations over the network via a Conditional AutoRegressive (CAR) model. We show that the conditional specification of the CAR model allows for a decentralized detection method to leverage spatial correlations by utilizing neighborhood information on each node. Theoretically, we prove that the expected detection delay for CARD is smaller than that for a detection method which ignores spatial correlations, thus showing the improved detection power of the proposed method. We then demonstrate the improved detection performance of CARD over existing methods in a suite of numerical simulations and in two applications: power grid monitoring, and sparse population coding of biological neural networks.

# CHAPTER 1

# DROPPING CONVEXITY FOR MORE EFFICIENT AND SCALABLE ONLINE MULTIVIEW LEARNING

## 1.1 Introduction

Multiview data have become increasingly available in many popular real-world data analysis and machine learning problems. These data are collected from diverse domains or different feature extractors, which share latent factors. Existing literature has demonstrated different scenarios. For instance, the pixels and captions of images can be considered as two-view data, since they are two different features describing the same contents. More motivating examples involving two or more data sets simultaneously can be found in computer vision, natural language processing, and acoustic recognition. See [1, 2, 3, 4, 5, 6, 7, 8]. Although these data are usually unlabeled, there exists underlying association and dependency between different views, which allows us to learn useful representations in an unsupervised manner. Here we are interested in finding a representation that reveals intrinsic low-dimensional structures and decomposes underlying confounding factors. One ubiquitous approach is partial least squares (PLS) for multiview representation learning. Specifically, given a data set of $n$ samples of two sets of random variables (views), $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^d$, PLS aims to find an $r$-dimensional subspace ($r \ll \min(m, d)$) that preserves most of the covariance between two views. Existing literature has shown that such a subspace is spanned by the leading $r$ components of the singular value decomposition (SVD) of $\Sigma_{XY} = \mathbb{E}_{(X,Y)\sim\mathcal{D}}\left[XY^\top\right]$ [9], where we sample $(X, Y)$ from some unknown distribution $\mathcal{D}$. Throughout the rest of this chapter, if not clear specified, we denote $\mathbb{E}_{(X,Y)\sim\mathcal{D}}$ by $\mathbb{E}$ for notational simplicity.

A straightforward approach for PLS is "Sample Average Approximation" (SAA, [10, 11]), where we run an offline (batch) SVD algorithm on the empirical covariance matrix after seeing sufficient data samples. However, in the "big data" regime, this approach requires unfeasible amount of storage and computation time. Therefore, it is much more practical to consider the multiview learning problem in a "data laden" setting, where we draw independent samples from an underlying distribution $\mathcal{D}$ over $\mathbb{R}^m \times \mathbb{R}^d$, one at a time. This further enables us to formulate PLS as a stochastic (online) optimization problem. Here we only consider the rank-1 case ($r = 1$) for simplicity, and solve

$$(\hat{u},\ \hat{v}) = \underset{u \in \mathbb{R}^m, v \in \mathbb{R}^d}{\operatorname{argmax}}\ \mathbb{E}\left(v^\top Y X^\top u\right) \quad \text{subject to} \quad u^\top u = 1, v^\top v = 1. \tag{1.1.1}$$

We will explain more details on the rank-$r$ case in the later section.

Several nonconvex stochastic approximation (SA) algorithms have been proposed in [9]. These algorithms work great in practice, but lack theoretic justifications, since the nonconvex nature of (1.1.1) makes the theoretical analysis very challenging. To overcome this obstacle, [12] proposed a convex relaxation of (1.1.1). Specifically, by a reparametrization $M = uv^\top$ (Recall that we are interested in the rank-1 PLS), they rewrite (1.1.1) as[1]

$$\hat{M} = \underset{M}{\operatorname{argmax}}\ \langle M, \Sigma_{XY} \rangle \quad \text{subject to} \quad \|M\|_* \leq 1 \text{ and } \|M\|_2 \leq 1. \tag{1.1.2}$$

where $\Sigma_{XY} = \mathbb{E}XY^\top$, and $\|M\|_2$ and $\|M\|_*$ are the spectral (*i.e.*, the largest singular value of $M$) and nuclear (*i.e.*, the sum of all singular values of $M$) norms of $M$ respectively. By examining the KKT conditions of (1.1.2), one can verify that $\hat{M} = \hat{u}\hat{v}^\top$ is the optimal solution, where $\hat{u}, \hat{v}$ are the leading left and right singular vectors of $\Sigma_{XY}$, *i.e.*, a pair of global optimal solutions to (1.1.1) for $r = 1$. Accordingly, they propose a projected stochastic gradient-type algorithm to solve (1.1.2), which is often referred to the Matrix

[1]For $r > 1$ case, we replace $\|M\|_* \leq 1$ with $\|M\|_* \leq r$

Stochastic Gradient (MSG) algorithm. Particularly, at the $(k+1)$-th iteration, MSG takes

$$M_{k+1} = \Pi_{\text{Fantope}}(M_k + \eta X_k Y_k^\top),$$

where $X_k$ and $Y_k$ are independently sampled from $\mathcal{D}$, and $\Pi_{\text{Fantope}}(\cdot)$ is a projection operator to the feasible set of (1.1.2). They further prove that given a pre-specified accuracy $\epsilon$, MSG requires $N = \mathcal{O}(\epsilon^{-2}\log(1/\epsilon))$ iterations such that $\langle \hat{M}, \mathbb{E}xy^\top \rangle - \langle M_N, \mathbb{E}xy^\top \rangle \leq \epsilon$ with high probability.

Despite of the attractive theoretic guarantee, MSG does not present superior performance to other heuristic nonconvex stochastic optimization algorithms for solving (1.1.1). Although there is a lack of theoretical justification, many evidences have corroborated that heuristic nonconvex approaches not only converge to the global optima in practice, but also enjoy better empirical computational performance than the convex approaches [13, 14, 15, 16]. Another drawback of MSG is the complicated projection step at each iteration. Although [12] further proposed an algorithm to compute the projection with a computational cost cubically depending on the rank of the iterates (the worst case: $\mathcal{O}(d^3)$), such a sophisticated implementation significantly decreases the practicability of MSG. Furthermore, MSG is also unfavored in a memory-restricted scenario, since storing the update $M^{(k)}$ requires $\mathcal{O}(md)$ real number storage. In contrast, the heuristic algorithms analyzed in this chapter require only $\mathcal{O}(m+d)$ real number storage, or $\mathcal{O}(rm+rd)$ in the rank-$r$ case.

We aim to bridge the gap between theory and practice for solving multiview representation learning problems by nonconvex approaches. Specifically, we first illustrate the nonoconvex geometry of (1.1.1), we analyze the convergence properties of a simple stochastic optimization algorithm for solving (1.1.1) based on diffusion processes. Our analysis takes advantage of the strong Markov properties of the stochastic optimization algorithm updates and casts the trajectories of the algorithms as diffusion processes [17, 18]. By leveraging the weak convergence from discrete Markov chains to their continuous time limits, we demonstrate that the trajectories are essentially the solutions to stochastic differential equations. Such an SDE-type analysis automatically incorporates the geometry of the objective and the randomness of the algorithm, and eventually demonstrates three phases of convergence:

1. Starting from an unstable equilibrium with negative curvature, the dynamics of the algorithm can be described by an Ornstein-Uhlenbeck process with a steady driven force pointing away from the initial.

2. When the algorithm is sufficiently distant from the initial unstable equilibrium, the dynamics can be characterized by a deterministic ordinary differential equation (ODE). The trajectory of this phase is evolving directly toward the desired global maximum until it reaches a small basin around the global maximum.

3. In this phase, the trajectory can be also described by an Ornstein-Uhlenbeck process oscillating around the global maximum. The process has a drifting term that gradually dies out and eventually becomes a nearly unbiased random walk centered at the maximum.

The sharp characterization in these three phases eventually allows us to establish strong convergence guarantees. Particularly, we show that the nonconvex stochastic gradient algorithm guarantees an $\epsilon$-optimal solution in $\mathcal{O}(\epsilon^{-1}\log(\epsilon^{-1}))$ iterations with high probability, which is a significant improvement over convex MSG by a factor of $\epsilon^{-1}$. Our theoretical analysis reveals the power of the nonconvex optimization in PLS. The simple heuristic algorithms drop the convexity, but achieve much better efficiency.

Our convergence analysis also has important implications on stochastic optimization algorithm for Canonical Correlation Analysis (CCA). Specifically, CCA considers a similar setting to PLS, and solves

$$(\hat{u},\hat{v}) = \underset{u,v}{\operatorname{argmax}}\ u^\top \mathbb{E}XY^\top v \quad \text{subject to} \quad \mathbb{E}(X^\top u)^2 = 1,\ \mathbb{E}(Y^\top v)^2 = 1. \tag{1.1.3}$$

From an optimization perspective, CCA is equivalent to PLS under some linear transformation, but more challenging. We will explain more details on CCA in our later discussions.

**Notations**: Given a vector $v = (v^{(1)}, \ldots, v^{(d)})^\top \in \mathbb{R}^d$, we define vector norms: $\|v\|_1 = \sum_j |v^{(j)}|$, $\|v\|_2^2 = \sum_j (v^{(j)})^2$, and $\|v\|_\infty = \max_j |v^{(j)}|$. Given a matrix $A \in \mathbb{R}^{d\times d}$, we use $A_j = (A_{1j}, ..., A_{dj})^\top$ to denote the $j$-th column of $A$ and define the matrix norms $\|A\|_{\mathrm{F}}^2 = \sum_j \|A_j\|_2^2$ and $\|A\|_2$ as the largest singular value of $A$.

### 1.2 Stochastic Nonconvex Optimization

Recall that we solve (1.1.1)

$$(\hat{u}, \hat{v}) = \underset{u,v}{\operatorname{argmax}}\ u^\top \mathbb{E}XY^\top v \quad \text{subject to} \quad \|u\|_2^2 = 1,\ \|v\|_2^2 = 1, \tag{1.2.1}$$

where $(X, Y)$ follows some unknown distribution $\mathcal{D}$. Due to the symmetrical structure of (1.2.1), $(-\hat{u}, -\hat{v})$ is also a pair of global optimum. Our analysis holds for both optima. Throughout the rest of this chapter, if not clearly specified, we consider $(\hat{u}, \hat{v})$ as the global optimum for simplicity.

We apply the stochastic approximation (SA) of the generalized Hebbian algorithm (GHA) to solve (1.2.1). GHA, which is also referred as Sanger's rule [19], is essentially a primal-dual algorithm. Specifically, we consider the Lagrangian function of (1.2.1):

$$L(u, v, \mu, \sigma) = u^\top \mathbb{E}XY^\top v - \mu(u^\top u - 1) - \sigma(v^\top v - 1), \tag{1.2.2}$$

where $\mu$ and $\sigma$ are Lagrangian multipliers. We then check the optimal KKT conditions,

$$\mathbb{E}XY^\top v - 2\mu u = 0, \quad \mathbb{E}YX^\top u - 2\sigma v = 0, \quad u^\top u = 1 \quad \text{and} \quad v^\top v = 1, \tag{1.2.3}$$

which further imply

$$u^\top \mathbb{E} XY^\top v - 2\mu u^\top u = u^\top \mathbb{E} XY^\top v - 2\mu = 0,$$

$$v^\top \mathbb{E} YX^\top u - 2\sigma v^\top v = v^\top \mathbb{E} YX^\top u - 2\sigma = 0.$$

Solving the above equations, we obtain the optimal Lagrangian multipliers as

$$\mu = \sigma = \frac{1}{2} u^\top \mathbb{E} XY^\top v. \tag{1.2.4}$$

GHA is inspired by (1.2.3) and (1.2.4). At $k$-th iteration GHA takes

$$\text{Dual Update :} \quad \mu_k \quad = \quad \sigma_k \quad = \quad \frac{1}{2} \underbrace{u_k^\top X_k Y_k^\top v_k}_{\text{SA (stochastic approximation) of } u_k^\top \Sigma v_k} \tag{1.2.5}$$

$$\begin{aligned} \text{Primal Update :} \quad u_{k+1} \quad &= \quad u_k + \eta \underbrace{\left( X_k Y_k^\top v_k - 2\mu_k u_k \right)}_{\text{SA of } \nabla_u L(u,v,\mu,\sigma)}, \\ v_{k+1} \quad &= \quad v_k + \eta \underbrace{\left( Y_k X_k^\top u_k - 2\sigma_k v_k \right)}_{\text{SA of } \nabla_v L(u,v,\mu,\sigma)}, \end{aligned} \tag{1.2.6}$$

where $\eta > 0$ is the step size. Combining (1.2.5) and (1.2.6), we obtain a dual-free update as follow:

$$\begin{aligned} u_{k+1} \ &= \ u_k + \eta \left( X_k Y_k^\top v_k - u_k^\top X_k Y_k^\top v_k u_k \right) \\ \text{and} \quad v_{k+1} \ &= \ v_k + \eta \left( Y_k X_k^\top u_k - u_k^\top X_k Y_k^\top v_k v_k \right). \end{aligned} \tag{1.2.7}$$

Different from the projected SGD algorithm, which is a primal algorithm proposed in [20], Stochastic GHA does not need projection at each iteration.

### 1.3 Optimization Landscape

We illustrate the nonconvex optimization landscape of (1.1.1), which helps us understand the intuition behind the algorithmic convergence. We first study its stationary points based the Lagrangian function (1.2.2). By the KKT conditions (1.2.3), we define the stationary point of (1.2.2) as follows.

**Definition 1.3.1.** Given (1.1.1) and (1.2.2), we define:

1. A quadruplet of $(u, v, \mu, \sigma)$ is called a stationary point of (1.2.2), if it satisfies (1.2.3).

2. A pair of $(u, v)$ is called a stable stationary point of (1.1.1), if $(u, v, \mu, \sigma)$ is a stationary point of (1.2.2), and $\nabla^2_{u,v} L(u, v, \mu, \sigma)$ is negative semi-definite.

3. A pair of $(u, v)$ is called an unstable stationary point of (1.1.1), if $(u, v, \mu, \sigma)$ is a stationary point of (1.2.2), and $\nabla^2_{u,v} L(u, v, \mu, \sigma)$ has a positive eigenvalue.

We then obtain all stationary points by solving (1.2.3). For notational simplicity, we denote $\Sigma_{XY} = \mathbb{E}XY^\top$. Before we proceed with our analysis, we introduce the following assumption.

**Assumption 1.3.2.** Suppose $d \leq m$ and rank$(\Sigma_{XY}) = r$. We have $\lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_r > 0$, where $\lambda_i$'s are the $i$-th singular values of $\Sigma_{XY}$.

We impose such an eigengap assumption ($\lambda_1 > \lambda_2$) to ensure the identifiability of the leading pair of singular vectors. Thus, the leading pair of singular vectors are uniquely determined only up to sign change. Let $O_1 \in \mathbb{R}^{m \times m}$ and $O_2 \in \mathbb{R}^{d \times d}$ be any pair of left and right singular matrices[2]. Let $\overline{u}_i$ and $\overline{v}_j$ denote the $i$-th column of $O_1$ and $j$-th column of $O_2$, respectively. The next proposition reveals the connection between stationary points and singular vectors.

**Proposition 1.3.3.** Suppose Assumption 1.3.2 holds. A quadruplet of $(u, v, \mu, \sigma)$ is the stationary point of (1.2.2), if either of the following condition holds:

1. $(u, v)$ are a pair of singular vectors associated with the same nonzero singular value;

[2]Since all singular values are not necessarily distinct, some pairs of singular vectors are not unique, *e.g.*, when $\lambda_i = \lambda_j$, $(\overline{u}_i, \overline{v}_i)$ and $(\overline{u}_j, \overline{v}_j)$ are uniquely determined up to rotation. Note that our analysis works for all possible combinations of $O_1$ and $O_2$. See more details in [21].

2. $u$ and $v$ belong to the row and column null spaces of $\Sigma_{XY}$ respectively: $\Sigma_{XY}v = 0,\ \Sigma_{XY}^\top u = 0$.

The proof of Proposition 1.3.3 is presented in Appendix A.1.1. We then determine the types of these obtained stationary points. The next proposition characterizes the maximum eigenvalues of $\nabla^2_{u,v}L(u,v,\mu,\sigma)$ at these stationary points of (1.2.2).

**Proposition 1.3.4.** Suppose Assumption 1.3.2 holds. All pairs of singular vectors associated with the leading singular value are global optima of (1.1.1), *i.e.*, also the saddle points of (1.2.2), and they are stable stationary points. All other stationary points of (1.2.2) are all unstable with

$$\lambda_{\max}(\nabla^2_{u,v}L(u,v,\mu,\sigma)) \geq \lambda_1 - \lambda_2.$$

The proof of Proposition 1.3.4 is presented in Appendix A.1.2. Proposition 1.3.4 essentially characterizes the geometry of (1.1.1) at all stationary points, and the unstableness allows the stochastic gradient algorithm to escape, as will be shown in the next sections.

### 1.4 Global Convergence by ODE

Before we proceed with our analysis, we first impose some mild assumptions on the problem.

**Assumption 1.4.1.** $X_k, Y_k,\ k = 1, 2, ...N$ are data samples identically independently distributed as $X \in \mathbb{R}^d,\ Y \in \mathbb{R}^d$ respectively satisfying the following conditions:

1. For any $\Delta > 0$, $\max\{\mathbb{E}\|X\|_2^{4+\Delta}, \mathbb{E}\|Y\|_2^{4+\Delta}\} < \infty$ and $\max\{\mathbb{E}\|X\|_2^2, \mathbb{E}\|Y\|_2^2\} \leq Bd$ for a constant $B$;[3]

2. $\lambda_1 > \lambda_2 \geq \lambda_3 \geq ... \geq \lambda_d > 0$, where $\lambda_i$'s are the singular values of $\Sigma_{XY} = \mathbb{E}XY^\top$.

[3]We only need $(4+\Delta)$-th moments of $\|X\|_2$ and $\|Y\|_2$ to be bounded, while the preliminary results in [20] require both $\|X\|_2$ and $\|Y\|_2$ to be bounded random variables.

Here we assume $X$ and $Y$ are of the same dimensions (***i.e.***, $m = d$) and $\Sigma_{XY}$ is full rank for convenience of analysis. The extension to $m \neq d$ in a rank deficient setting is straightforward, but more involved (See more details in Section 1.5.4). Moreover, for a multiview learning problem, it is also natural to impose the following additional assumptions.

**Assumption 1.4.2.** Given the observed random variables $X$ and $Y$, there exist two orthogonal matrices $O_X \in \mathbb{R}^{d\times d}$, $O_Y \in \mathbb{R}^{d\times d}$ such that $X = O_X\overline{X}$, $Y = O_Y\overline{Y}$, where $\overline{X} = (\overline{X}^{(1)}, ..., \overline{X}^{(d)})^\top \in \mathbb{R}^d$ and $\overline{Y} = (\overline{Y}^{(1)}, ..., \overline{Y}^{(d)})^\top \in \mathbb{R}^d$ are the latent variables satisfying:

1. $\overline{X}^{(i)}$ and $\overline{Y}^{(j)}$ are uncorrelated if $i \neq j$, so that $O_X$ and $O_Y$ are the left and right singular matrices of $\Sigma_{XY}$ respectively;
2. $\mathrm{Var}(\overline{X}^{(i)}) = \gamma_i, \quad \mathrm{Var}(\overline{Y}^{(i)}) = \omega_i, \;\; \mathbb{E}\left(\overline{X}^{(i)}\overline{Y}^{(i)}\overline{X}^{(j)}\overline{Y}^{(j)}\right) = \alpha_{ij}$, where $\gamma_i, \alpha_{ij}$, and $\omega_i$ are constants.

The next proposition characterizes the strong Markov property of our algorithm.

**Proposition 1.4.3.** Using (1.2.7), we get a sequence of $(u_k, v_k),\; k = 1, 2, ..., N$. They form a discrete-time Markov process.

With Proposition 1.4.3, we can construct a continuous time process to derive an ordinary differential equation to analyze the algorithmic convergence. Specifically, as the fixed step size $\eta \to 0^+$, two processes $U_\eta(t) = u_{\lfloor \eta^{-1}t\rfloor},\; V_\eta(t) = v_{\lfloor \eta^{-1}t\rfloor}$ based on the sequence generated by (1.2.7), weakly converge to the solution of the following ODE system in probability (see more details in [17]),

$$\frac{dU}{dt} = \left(\Sigma_{XY}V - U^\top\Sigma_{XY}VU\right), \quad \frac{dV}{dt} = \left(\Sigma_{XY}^\top U - V^\top\Sigma_{XY}^\top UV\right), \tag{1.4.1}$$

where $U(0) = u_0$ and $V(0) = v_0$. To highlight the sequence generated by (1.2.7) depending on $\eta$, we redefine $u_{\eta,k} = u_k,\; v_{\eta,k} = v_k$.

**Theorem 1.4.4.** As $\eta \to 0^+$, the processes $u_{\eta,k},\ v_{\eta,k}$ weakly converge to the solution of the ODE system in (1.4.1) with sphere initial $U(0) = u_0,\ V(0) = v_0$, *i.e.*, $\|u_0\|_2 = \|v_0\|_2 = 1$.

The proof of Theorem 1.4.4 is presented in Appendix A.2.1. Under Assumption 1.4.1, the above ODE system admits a closed form solution. Specifically, we solve $U$ and $V$ simultaneously, since they are coupled together in (1.4.1). To simplify (1.4.1), we define $W = \frac{1}{\sqrt{2}}\left(U^\top\ V^\top\right)^\top$ and $w_k = \frac{1}{\sqrt{2}}\left(u_k^\top\ v_k^\top\right)^\top$. We then rewrite (1.4.1)as

$$\frac{dW}{dt} = QW - W^\top QWW, \tag{1.4.2}$$

where $Q = \begin{pmatrix} 0 & \Sigma_{XY} \\ \Sigma_{XY}^\top & 0 \end{pmatrix}$. By Assumption 1.4.2, $O_X$ and $O_Y$ are the left and right singular matrices of $\Sigma_{XY}$ respectively, *i.e.*, $\Sigma_{XY} = \mathbb{E}XY^\top = O_X \mathbb{E}\overline{XY}^\top O_Y^\top$, where $\mathbb{E}\overline{XY}^\top$ is diagonal. For notational simplicity, we define $D = \operatorname{diag}(\lambda_1, \lambda_2, ..., \lambda_d)$ such that $\Sigma_{XY} = O_X D O_Y^\top$. One can verify $Q = P\Lambda P^\top$, where

$$P = \frac{1}{\sqrt{2}}\begin{pmatrix} O_X & O_X \\ O_Y & -O_Y \end{pmatrix} \quad \text{and} \quad \Lambda = \begin{pmatrix} D & 0 \\ 0 & -D \end{pmatrix}. \tag{1.4.3}$$

By left multiplying $P^\top$ both sides of (1.4.2), we obtain

$$H(t) = P^\top W(t) \text{ with } \frac{dH}{dt} = \Lambda H - H^\top \Lambda H H, \tag{1.4.4}$$

which is a coordinate separable ODE system. Accordingly, we define $h_k^{(i)}$'s as:

$$h_k = P^\top w_k \quad \text{and} \quad h_k^{(i)} = P_i^\top w_k. \tag{1.4.5}$$

Thus, we can obtain a closed form solution to (1.4.4) based on the following theorem.

**Theorem 1.4.5.** Given (1.4.4), we write the ODE in each component $H^{(i)}$,

$$\frac{d}{dt} H^{(i)} = H^{(i)} \sum_{j=1}^{2d} (\lambda_i - \lambda_j) \, (H^{(j)})^2, \tag{1.4.6}$$

where $\lambda_i = -\lambda_{i-d}$ when $i > d$. This ODE System has a closed form solution as follows:

$$H^{(i)}(t) = \big(C(t)\big)^{-\frac{1}{2}} H^{(i)}(0) \exp{(\lambda_i t)}, \tag{1.4.7}$$

for $i = 1, 2, ..., 2d$, where

$$C(t) = \sum_{j=1}^{2d} \Big( \big(H^{(j)}(0)\big)^2 \exp{(2\lambda_j t)} \Big)$$

is a normalization function such that $\|H(t)\|_2 = 1$.

The proof of Theorem 1.4.5 is presented in Appendix A.2.2. Without loss of generality, we assume $H^{(1)}(0) > 0$. As can be seen, $H_1(t) \to 1$, as $t \to \infty$. We have successfully characterized the global convergence performance of our algorithm with an approximate error $o(1)$. The solution to the ODE system in (1.4.7), however, does not fully reveal the algorithmic behavior (more precisely, the rate of convergence) near the equilibria of the ODE system. This further motivates us to exploit the stochastic differential equation approach to characterize the dynamics of the algorithm.

## 1.5 Global Dynamics by SDE

We analyze the dynamics of the algorithm near the equilibria based on stochastic differential equation by rescaling analysis. Specifically, we characterize three stages for the trajectories of solutions: [a] Neighborhood around unstable equilibria — minimizers and saddle points of (1.2.1), [b] Neighborhood around stable equilibria — maximizers of (1.2.1), and [c] deterministic traverses between equilibria. Moreover, we provide the approximate the number of iterations in each phase until convergence.

### 1.5.1 Phase I: Escaping from Unstable Equilibria

Suppose that the algorithm starts to iterate around a unstable equilibrium, (*e.g.* saddle point). Different from our previous analysis, we rescale two aforementioned processes $U_\eta(t)$ and $V_\eta(t)$ rescaled by a factor of $\eta^{-1/2}$. This eventually allows us to capture the uncertainty of the algorithm updates by stochastic differential equations. Roughly speaking, the ODE approximation is essentially a variant of law of large number for Markov process, while the SDE approximation serves as a variant of central limit theorem accordingly.

Recall that $P$ is an orthonormal matrix for diagonalizing Q, and $H$ is defined in (1.4.4). Let $Z_\eta^{(i)}$ and $z_{\eta,k}^{(i)}$ denote the $i$-th coordinates of $Z_\eta = \eta^{-1/2}H_\eta$ and $z_{\eta,k} = \eta^{-1/2}h_{\eta,k}$ respectively. The following theorem characterizes the dynamics of the algorithm around the unstable equilibrium.

**Theorem 1.5.1.** Suppose $z_{\eta,0}$ is initialized around some saddle point or minimizer (*e.g.* $j$-th column of $P$ with $j \neq 1$), i.e., $Z^{(j)}(0) \approx \eta^{-\frac{1}{2}}$ and $Z^{(i)}(0) \approx 0$ for $i \neq j$. Then for any $C > 0$, there exist $\tau > 0$ and $\eta' > 0$ such that

$$\sup_{\eta<\eta'} \mathbb{P}(\sup_t |Z_\eta^{(i)}(t)| \leq C) \leq 1 - \tau. \tag{1.5.1}$$

Here we provide the proof sketch and leave the whole proof of Theorem 1.5.1 in Appendix A.3.1.

*Proof Sketch.* We prove this argument by contradiction. Assume the conclusion does not hold, that is there exists a constant $C > 0$, such that for any $\eta' > 0$ we have

$$\sup_{\eta\leq\eta'} \mathbb{P}(\sup_t |Z_\eta^{(i)}(t)| \leq C) = 1.$$

That implies there exists a sequence $\{\eta_n\}_{n=1}^{\infty}$ converging to $0$ such that

$$\lim_{n\to\infty} \mathbb{P}(\sup_t |Z_{\eta_n}^{(i)}(t)| \le C) = 1. \tag{1.5.2}$$

Then we show $\{Z_{\eta_n}^{(i)}(\cdot)\}_n$ is tight and thus converges weakly. Furthermore, $\{Z_{\eta_n}^{(i)}(\cdot)\}_n$ weakly converges to a stochastic differential equation,

$$dZ^{(i)}(t) = -(\lambda_j - \lambda_i)Z^{(i)}(t)dt + \beta_{ij}dB(t). \tag{1.5.3}$$

We compute the solution of this stochastic differential equation and then show (2.4.6) does not hold. $\square$

Theorem 1.5.1 implies that for $i > j$, with a constant probability $\tau$, escapes from the saddle points at some time $T_1$, i.e., $(H^{(j)}(T_1))^2$ is smaller than $1 - \delta^2$, where $(\delta = O(\sqrt{\eta}))$. Note that (1.5.3) is a Fokker-Planck equation, which admits a closed form solution as follows, for any $i \neq j$,

$$\begin{aligned} Z^{(i)}(t) &= Z^{(i)}(0)\exp\left[-(\lambda_j - \lambda_i)t\right] + \beta_{ij}\int_0^t \exp\left[(\lambda_j - \lambda_i)(s-t)\right]dB(s) \\ &= \Bigg[\underbrace{Z^{(i)}(0) + \beta_{ij}\int_0^t \exp\left[(\lambda_j - \lambda_i)s\right]dB(s)}_{T_1}\Bigg]\underbrace{\exp\left[(\lambda_i - \lambda_j)t\right]}_{T_2}. \end{aligned} \tag{1.5.4}$$

Such a solution is well known as the Ornstein-Uhlenbeck process [22], and also implies that the distribution of $z_{\eta,k}^{(i)}$ can be well approximated by the normal distribution of $Z^{(i)}(t)$ for a sufficiently small step size. This continuous approximation further has the following implications:

**[a]** For $\lambda_i > \lambda_j,\ T_1 = \beta_{ij}\int_0^t \exp\big[(\lambda_j - \lambda_i)s\big]dB(s) + Z^{(i)}(0)$ is essentially a random variable with mean $Z^{(i)}(0)$ and variance smaller than $\frac{\beta_{ij}^2}{2(\lambda_i - \lambda_j)}$. The larger $t$ is, the closer its variance gets to this upper bound. While $T_2 = \exp\big[(\lambda_i - \lambda_j)t\big]$ essentially amplifies $T_1$ by a factor exponentially increasing in $t$. This tremendous amplification forces $Z^{(i)}(t)$ to quickly get away from $0$, as $t$ increases.

**[b]** For $\lambda_i < \lambda_j$, we have

$$\mathbb{E}[Z^{(i)}(t)] = Z^{(i)}(0)\exp\left[-(\lambda_j - \lambda_i)t\right]$$
$$\text{and}\quad \mathrm{Var}[Z^{(i)}(t)] = \frac{\beta_{ij}^2}{2(\lambda_j - \lambda_i)}\big[1 - \exp\left[-2(\lambda_j - \lambda_i)t\right]\big].$$

As has been shown in **[a]** that $t$ does not need to be large for $Z^{(i)}(t)$ to get away from $0$. Here we only consider relatively small $t$. Since the initial drift for $Z^{(i)}(0) \approx 0$ is very small, $Z^{(i)}$ tends to stay at $0$. As $t$ increases, the exponential decay term makes the drift quickly become negligible. Moreover, by mean value theorem, we know that the variance is bounded, and increases far slower than the variance in **[a]**. Thus, roughly speaking, $Z^{(i)}(t)$ oscillates near $0$.

**[c]** For $\lambda_j = \lambda_i$, we have $\mathbb{E}[Z^{(i)}(t)] = Z^{(i)}(0)$ and $\mathrm{Var}[Z^{(i)}(t)] = \beta_{ij}^2$. This implies that $Z^{(i)}(t)$ also tends to oscillate around $0$, as $t$ increases.

Overall speaking, **[a]** is dominative so that it is the major driving force for the algorithm to escape from this unstable equilibrium. More precisely, let us consider one special case for Phase I, that is we start from the second maximum singular value, with $h_{\eta,k}^{(2)}(0) = 1$. We then approximately calculate the number of iterations to escape Phase I using the algorithmic behavior of $h_{\eta,k}^{(1)} = \eta^{1/2} z_{\eta,k}^{(1)} \approx \eta^{1/2} Z_\eta^{(1)}(t)$ with $t = k\eta$ by the following proposition.

**Proposition 1.5.2.** Given pre-specified $\nu > 0$ and sufficiently small $\eta$, there exists some $\delta \asymp \eta^\mu$, where $\mu \in (0, 0.5)$ is a generic constant, such that the following result holds: We need at most

$$N_1 = \frac{\eta^{-1}}{2(\lambda_1 - \lambda_2)} \log\left(\frac{2\eta^{-1}\delta^2(\lambda_1 - \lambda_2)}{\Phi^{-1}\left(\frac{1+\nu}{2}\right)^2 \beta_{12}^2} + 1\right)$$

iterations such that $(h_{\eta,N_1}^{(2)})^2 \leq 1 - \delta^2$ with probability at least $1 - \nu$, where $\Phi(x)$ is the CDF of standard normal distribution.

The proof of Proposition 1.5.2 is provided in Appendix A.3.2. Proposition 1.5.2 suggests that SGD can escape from unstable equilibria within a few iterations. After escaping from the saddle, SGD gets into the next phase, which is a deterministic traverse between equilibria.

### 1.5.2 Phase II: Traverse between Equilibria

When the algorithm is close to neither the saddle points nor the optima, the algorithm's performance is nearly deterministic. Since $Z(t)$ is a rescaled version of $H(t)$, their trajectories are similar. Like before, we have the following proposition to calculate the approximate iterations, $N_2$, following our results in Section 1.4. We restart the counter of iteration by Proposition 1.4.3.

**Proposition 1.5.3.** After restarting counter of iteration, given sufficiently small $\eta$ and $\delta$ defined in Proposition 1.5.2, we need at most

$$N_2 = \frac{\eta^{-1}}{2(\lambda_1 - \lambda_2)} \log \frac{1-\delta^2}{\delta^2}$$

iterations such that $\left(h_{\eta,N_2}^{(1)}\right)^2 \geq 1 - \delta^2$.

The proof of Proposition 1.5.3 is provided in Appendix A.3.3. Combining Propositions 1.5.2 and 1.5.3, we know that after $N_1 + N_2$ iteration numbers, SGD is close to the optimum with high probability, and gets into its third phase, i.e., convergence to stable equilibria.

### 1.5.3 Phase III: Convergence to Stable Equilibria

Again, we restart the counter of iteration by the strong Markov property. The trajectory and analysis are similar to Phase I, since we also characterize the convergence using an Ornstein-Uhlenbeck process. The following theorem characterizes the dynamics of the algorithm around the stable equilibrium.

**Theorem 1.5.4.** Suppose $z_{\eta,0}$ is initialized around some maximizer (the first column of $P$), i.e., $Z^{(1)}(0) \approx \eta^{-\frac{1}{2}}$ and $Z^{(i)}(0) \approx 0$ for $i \neq 1$. Then as $\eta \to 0^+$, for all $i \neq 1$, $z_{\eta,k}^{(i)}$ weakly converges to a diffusion process $Z^{(i)}(t)$ satisfying the following SDE for $i \neq 1$,

$$dZ^{(i)}(t) = -(\lambda_1 - \lambda_i)Z^{(i)}(t)dt + \beta_{i1}dB(t), \tag{1.5.5}$$

where $B(t)$ is a brownian motion, and

$$\beta_{i1} = \begin{cases} \frac{1}{2}\sqrt{\gamma_i\omega_1 + \gamma_1\omega_i + 2\alpha_{i1}} & \text{if } 1 \leq i \leq d, \\ \frac{1}{2}\sqrt{\gamma_i\omega_1 + \gamma_1\omega_i - 2\alpha_{i1}} & \text{otherwise.} \end{cases}$$

The proof of Theorem 1.5.4 is provided in Appendix A.3.4. Similar to (1.5.4), the closed form solution to (1.5.5) for $i \neq 1$ is as follow:

$$Z^{(i)}(t) = Z^{(i)}(0)\exp\left[-(\lambda_1 - \lambda_i)t\right] + \beta_{i1}\int_0^t \exp\left[(\lambda_1 - \lambda_i)(s-t)\right]dB(s). \tag{1.5.6}$$

By the property of the O-U process, we characterize the expectation and variance of $Z^{(i)}(t)$ for $i \neq 1$:

$$\begin{aligned} &\mathbb{E}Z^{(i)}(t) = Z^{(i)}(0)\exp\left[-(\lambda_1 - \lambda_i)t\right], \\ &\mathbb{E}\left(Z^{(i)}(t)\right)^2 = \frac{\beta_{i1}^2}{2(\lambda_1 - \lambda_i)} + \left[\left(Z^{(i)}(0)\right)^2 - \frac{\beta_{i1}^2}{2(\lambda_1 - \lambda_i)}\right]\exp\left[-2(\lambda_1 - \lambda_i)t\right]. \end{aligned}$$

Recall that the distribution of $z_{\eta,k}^{(i)}$ can be well approximated by the normal distribution of $Z^{(i)}(t)$ for a sufficiently small step size. This further implies that after sufficiently many iterations, SGD enforces $z_{\eta,k}^{(i)} \to 0$ except $i = 1$. Meanwhile, SGD behaves like a biased random walk towards the optimum, when it iterates within a small neighborhood the optimum. But unlike Phase I, the variance gradually becomes a constant.

Based on Theorem 1.5.4, we further establish an iteration complexity bound for SGD in following proposition.

**Proposition 1.5.5.** Given a pre-specified $\epsilon > 0$, a sufficiently small $\eta$, and $\delta$ defined in Proposition 1.5.2, after restarting counter of iteration, we need at most

$$N_3 = \frac{\eta^{-1}}{2(\lambda_1 - \lambda_2)} \log\left(\frac{4(\lambda_1 - \lambda_2)\delta^2}{(\lambda_1 - \lambda_2)\epsilon\eta^{-1} - 4d \max\limits_{1\le i\le d} \beta_{i1}^2}\right)$$

iterations such that $\sum_{i=2}^{2d}\left(h_{\eta,N_3}^{(i)}\right)^2 \le \epsilon$ with probability at least $3/4$.

The proof of Proposition 1.5.5 is provided in Appendix A.3.5. Combining Propositions 1.5.2, 1.5.3, and 1.5.5, we obtain a more refined result in the following corollary.

**Corollary 1.5.6.** Given a sufficiently small pre-specified $\epsilon > 0$, we choose

$$\eta \asymp \frac{\epsilon(\lambda_1 - \lambda_2)}{d \max_{1\le i\le d} \beta_{i1}^2}.$$

We need at most

$$N = O\left[\frac{d}{\epsilon(\lambda_1 - \lambda_2)^2}\log\left(\frac{d}{\epsilon}\right)\right]$$

iterations such that we have $\|u_{\eta,n} - \hat{u}\|_2^2 + \|v_{\eta,n} - \hat{v}\|_2^2 \le 3\epsilon$ with probability at least $\frac{3}{4}$.

The proof of Corollary 1.5.6 is provided in Appendix A.3.6. We can further improve the probability to $1 - \nu$ for some $\nu > 0$ by repeating $\mathcal{O}(\log 1/\nu)$ replicates of SGD. We then compute the geometric median of all output solutions. See more details in [23].

### 1.5.4 Extension to $m \neq d$

Our analysis can further extend to the case where $X$ and $Y$ have different dimensions, i.e., $m \neq d$. Specifically, we consider an alternative way to construct $P$ defined in (1.4.3). We follow the same notations to Assumption 1.4.2, and use $O_X$ and $O_Y$ to denote the transition matrix between the observed data and latent variables. The dimensions of $O_X$ and $O_Y$, however, are different now, i.e., $O_X \in \mathbb{R}^{m\times m}$ and $O_Y \in \mathbb{R}^{d\times d}$. Without loss of generality, we assume $m > d$ and $O_X = (\tilde{O}_X \; O_X^0)$, where $\tilde{O}_X \in \mathbb{R}^{m\times d}$ and $O_X^0 \in \mathbb{R}^{m\times(m-d)}$, and $O_Y$ are the transform matrix of $X$ and $Y$, respectively. Then we have the singular value decomposition as follows,

$$O_X^\top \Sigma_{XY} O_Y = D, \quad \text{where } D = \begin{pmatrix} \tilde{D} \\ 0 \end{pmatrix} \text{ and } \tilde{D} = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_d). \tag{1.5.7}$$

Thus, we have $\tilde{O}_X^\top \Sigma_{XY} O_Y = \tilde{D}$ and $(O_X^0)^\top \Sigma_{XY} O_Y = 0$. Now we design the orthogonal transform matrix $P$.

$$P = \begin{pmatrix} \frac{1}{\sqrt{2}}\tilde{O}_X & O_X^0 & \frac{1}{\sqrt{2}}\tilde{O}_X \\ \frac{1}{\sqrt{2}}O_Y & 0 & -\frac{1}{\sqrt{2}}O_Y \end{pmatrix}. \tag{1.5.8}$$

One can check that

$$\begin{pmatrix} 0 & \Sigma_{XY} \\ \Sigma_{XY}^\top & 0 \end{pmatrix} = P \begin{pmatrix} D & 0 \\ 0 & -D^\top \end{pmatrix} P^\top = P \begin{pmatrix} \tilde{D} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\tilde{D} \end{pmatrix} P^\top. \tag{1.5.9}$$

Then our previous analysis using ODE and SDE still holds.

Note that for $d = m$, any column vector of $P$ in (1.4.3) is a stationary solution. Here the square matrix $P$ in (1.5.8) contains $m + d$ column vectors, but only the first $d$ and last $d$ column vectors are stationary solutions. This is because the remaining $m - d$ column vectors are even not feasible solutions, and violate the constraint $v^\top v = 1$. Thus, given a feasible initial, the algorithm will not be trapped in the subspace spanned by th remaining $m - d$ column vectors .

### 1.5.5 Extension to Missing Values

Our methodology and theory can tolerate missing values. For simplicity, we assume the entries of $X$ and $Y$ misses independently with probability $1 - p$ in each iteration, where

$p \in (0,1)$. We then set all missing entries as $0$ values. We denote such imputed vectors by $\tilde{X}_k$ and $\tilde{Y}_k$. One can verify $\frac{1}{p^2}\tilde{X}_k \cdot \tilde{Y}_k^\top$ is an unbiased estimator of $\Sigma_{XY} = \mathbb{E}X_k Y_k^\top$. Note that $1/p^2$ can be further absorbed into the step size $\eta$, denoted by $\eta_p$. Then (1.2.7) becomes:

$$
\begin{aligned}
& u_{k+1} = u_k + \eta_p \left( \tilde{X}_k \tilde{Y}_k{}^\top v_k - u_k^\top \tilde{X}_k \tilde{Y}_k{}^\top v_k u_k \right) \\
\text{and} \quad & v_{k+1} = v_k + \eta_p \left( \tilde{Y}_k \tilde{X}_k{}^\top u_k - u_k^\top \tilde{X}_k \tilde{Y}_k{}^\top v_k v_k \right).
\end{aligned} \tag{1.5.10}
$$

The convergence analysis is very similar to the standard setting with a different choice of $\eta_p$, and therefore is omitted.

### 1.6 Numerical Experiments

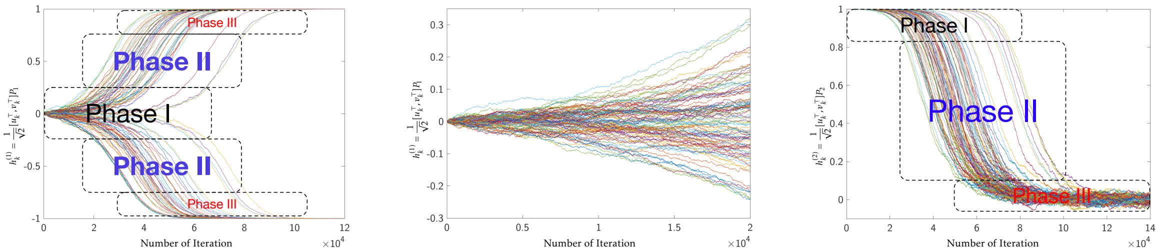
We first provide a simple example to illustrate our theoretical analysis. Specifically, we choose $m = d = 3$. We first generate the joint covariance matrix for the latent factors $\overline{X}$ and $\overline{Y}$ as

$$
\mathrm{Cov}(\overline{X}) = \Sigma_{\overline{XX}} = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 6 & 2 \\ 1 & 2 & 6 \end{bmatrix}, \quad \mathrm{Cov}(\overline{X}, \overline{Y}) = \Sigma_{\overline{XY}} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0.5 \end{bmatrix},
$$

and $\Sigma_{\overline{YY}} = \Sigma_{\overline{XX}}$. We then generate two matrices $\tilde{U}$ and $\tilde{V}$ with each entry independently sampled from $N(0,1)$. Then we convert $\tilde{U}$ and $\tilde{V}$ to orthonormal matrices $U$ and $V$ by Grand-Schmidt transformation. At last, we generate the joint covariance matrix for the observational random vectors $X$ and $Y$ using the following covariance matrix

$$
\mathrm{Cov}(X) = U^\top \Sigma_{\overline{XX}} U, \quad \mathrm{Cov}(X,Y) = U^\top \Sigma_{\overline{XY}} V, \quad \text{and} \quad \mathrm{Cov}(Y) = V^\top \Sigma_{\overline{YY}} V.
$$

We consider the total sample size as $n = 2 \times 10^5$ and choose $\eta = 5 \times 10^{-5}$. The initialization solution $(u_0, v_0)$ is a pair of singular vectors associated with the second largest singular




(a) All Three Phases of $h_k^{(1)}$. (b) Phase I of $h_k^{(1)}$. (c) All Three Phases of $h_k^{(2)}$.

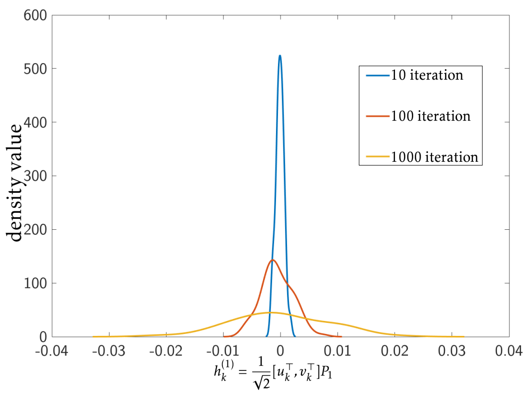
Figure 1.1: An illustrative example of the stochastic gradient algorithm. The three phases of the algorithm are consistent with our theory: In Phase I, the algorithm gradually escapes from the saddle point; In Phase II, the algorithm quickly iterates towards the optimum; In Phase III, the algorithm gradually converges to the optimum.

value of $\Sigma_{XY}$, *i.e.*, saddle point. We repeat the simulation with update (1.2.7) for $100$ times, and plot the obtained results.

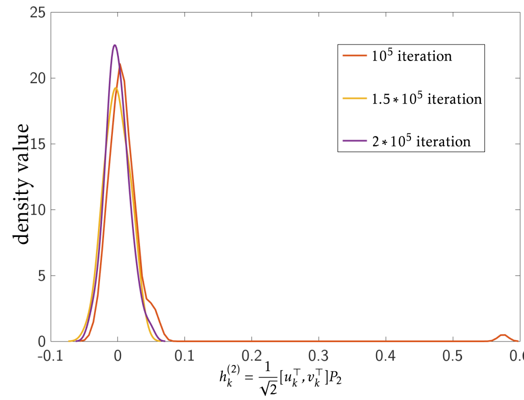
Figure 1.1a illustrates the three phases of the SGD algorithm. Specifically, the horizontal axis is the number of iterations, and the vertical axis is $h_k^{(1)}$ defined in (1.4.5). As $h_k^{(1)} \to \pm 1$, we have $u_k \to \pm \hat{u}$ and $v_k \to \pm \hat{v}$, *e.g.*, global optima. This is due to the symmetric structure of the problem as mentioned in Section 1.1. Figure 1.1a is consistent with our theory: In Phase I, the algorithm gradually escapes from the saddle point; In Phase II, the algorithm quickly moves towards the optimum; In Phase III, the algorithm gradually converges to the optimum.

Figure 1.1b further zooms in Phase I of Figure 1.1a. We see that the trajectories of all $100$ simulations behave very similar to an O-U process. Figure 1.1c illustrates the three phases by $h_k^{(2)}$. As our analysis suggests, when $h_k^{(1)} \to \pm 1$, we have $h_k^{(2)} \to 0$. We see that the trajectories of all $100$ simulations also behave very similar to an O-U process in Phase III. These experimental results are consistent with our theory.

Also, we illustrate $h^{(1)}$ in Phase I and $h^{(2)}$ in Phase III are O-U processes by showing that $100$ simulations of $h^{(1)}$ follow gaussian distributions at $10$-th, $100$-th, and $1000$-th iteration and those of $h^{(1)}$ follow gaussian distributions at $10^5$-th, $1.5 \times 10^5$-th, and $2 \times 10^5$-th iteration. This is consistent with the Theorems 1.5.1 and 1.5.4 in Section 1.5. Also as we can see that in the Phase I, the variance of $h^{(1)}$ becomes larger and larger when the iteration number increases. Similarly, in the Phase III, the variance of $h^{(2)}$ becomes closer to a fixed number.




(a) The estimated density of $h^{(1)}$ in Phase I.

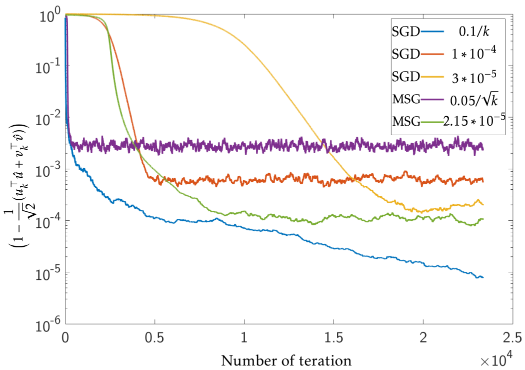



(b) The estimated density of $h^{(2)}$ in Phase III.

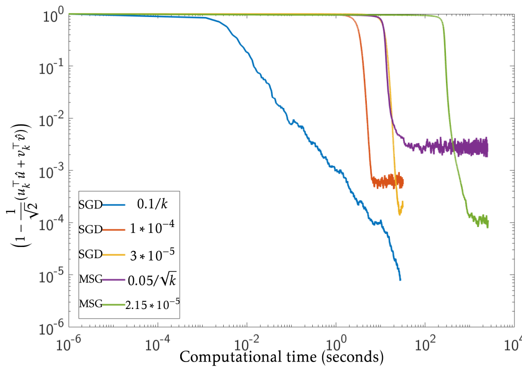
Figure 1.2: The estimated density based on 100 simulations (obtained by kernel density estimation using 10-fold cross validation) at different iterations in Phase I and Phase III shows that $h_k^{(1)}$'s in Phase I and $h_k^{(2)}$'s in Phase III behave very similar to O-U processes. how their their variance change, which is consistent our theory.

We then provide a real data experiment for comparing the computational performance our nonconvex stochastic gradient algorithm for solving (1.2.1) with the convex stochastic gradient algorithm for solving (1.1.2). We choose a subset of the MNIST dataset, whose labels are $3, 4, 5,$ or $9$. The total sample size is $n = 23343$, and $m = d = 392$. As [12] suggest, we choose $\eta_k = 0.05/\sqrt{k}$ or $2.15 \times 10^{-5}$, for the convex stochastic gradient algorithm. For our nonconvex stochastic gradient algorithm, we choose either $\eta_k = 0.1/k$, $10^{-4}$, or $3 \times 10^{-5}$. Figure 1.3 illustrates the computational performance in terms of iterations and wall clock time. As can be seen, our nonconvex stochastic gradient algorithm outperforms the convex counterpart in iteration complexity, and significantly outperforms in wall clock time, since the nonconvex algorithm does not need the computationally expensive projection in each iteration. This suggests that dropping convexity for PLS can boost both computational scalability and efficiency.

Our last experiment demonstrates the computational performance of our proposed SGD algorithm when there exist missing values. Specifically, we adopt the same MNIST data

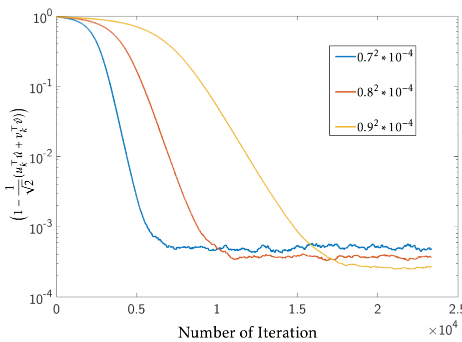



(a) Comparison by Iteration.

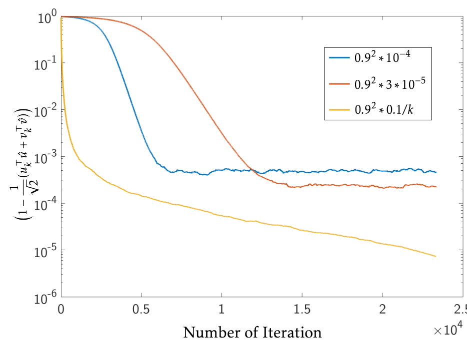



(b) Comparison by Time.

Figure 1.3: Comparison between nonconvex SGD and convex MSG with different step sizes. We see that SGD not only has a better iteration complexity, but also is more computationally efficient in wall clock time than convex MSG.




(a) Different missing probabilities with step size $p^2 * 10^{-4}$.




(b) Different step sizes with missing probability $0.1$ (*i.e.*, $p = 0.9$).

Figure 1.4: Comparison among different missing probabilities and step sizes.

set as our previous experiment. We independently drop each pixel of the image in each iteration with probability $(1 - p)$. Figure 1.4 illustrates the computational performance in terms of iterations under different missing probability and choices of the step size parameter. As can be seen, the empirical convergence of our proposed SGD algorithm is similar to (but slower than) that of our previous experiment without missing values.

## 1.7 Discussions

We establish the convergence rate of stochastic gradient descent (SGD) algorithms for solving online partial least square (PLS) problems based on diffusion process approximation. Our analysis indicates that for PLS, dropping convexity actually improves efficiency and scalability. Our convergence results are tighter than existing convex relaxation based method by a factor of $O(1/\epsilon)$, where $\epsilon$ is a pre-specified error. We believe the following directions should be of wide interests:

1. Our current results hold only for the top pair of left and right singular vectors, i.e., $r = 1$. For $r > 1$, we need to solve

$$\begin{aligned} (\hat{U},\ \hat{V}) = & \underset{U \in \mathbb{R}^{m\times r}, V \in \mathbb{R}^{d\times r}}{\operatorname{argmax}} \quad \mathbb{E}\ \mathrm{tr}(V^\top Y X^\top U) \\ \text{subject to} & \quad U^\top U = I_r, \quad V^\top V = I_r. \end{aligned} \tag{1.7.1}$$

   Our approximations using ODE and SDE, however, do not admit unique solution due to rotation or permutation. Thus, extension of our analysis to $r > 1$ is a challenging, but also an important future direction.

2. Our current results are only applicable to a fixed step size $\eta \asymp \epsilon(\lambda_1 - \lambda_2)d^{-1}$. Our experiments suggest that the diminishing step size $\eta_k \asymp k^{-1}(\lambda_1 - \lambda_2)^{-1}\log d$, $k$ from 1 to $N$, where $N$ is the sample complexity from theory, achieves a better empirical performance. One possible probability tool is Stein's method [24].

3. Our current results rely on the classical central limit theorem-type analysis by taking $\eta \to 0^+$. Note the analysis of $\|u\|_2 = \|v\|_2 = 1$ is an asymptotic result, and in experiment, when $\eta$ is small, $u$ and $v$ exactly stay on the sphere. But to get a more general result, connecting our analysis to discrete algorithmic proofs such as [25, 26, 27] should be an important direction [28]. One possible probability tool for addressing this issue is Stein's method [24].

Moreover, our proposed SGD algorithm for PLS is also closely related to Canonical Correlation Analysis. Specifically, CCA solves a similar problem

$$(\hat{u}, \hat{v}) = \underset{u,v}{\operatorname{argmax}}\ u^\top \mathbb{E} XY^\top v \quad \text{subject to} \quad \mathbb{E}(X^\top u)^2 = 1,\ \mathbb{E}(Y^\top v)^2 = 1. \tag{1.7.2}$$

For notational simplicity, we denote $\Sigma_{XY} = \mathbb{E}XY^\top$, $\Sigma_{XX} = \mathbb{E}XX^\top$, and $\Sigma_{YY} = \mathbb{E}YY^\top$. Since computing $\mathbb{E}XX^\top$ and $\mathbb{E}YY^\top$ is not affordable, the projected stochastic gradient algorithms are not applicable. Thus we consider an alternative approach to avoid the projection operation. We consider the Lagrangian function of (1.7.2) as

$$L(u, v, \mu, \sigma) = u^\top \Sigma_{XY} v - \mu(u^\top \Sigma_{XX} u - 1) - \sigma(v^\top \Sigma_{YY} v - 1), \tag{1.7.3}$$

where $\mu$ and $\sigma$ are Lagrangian multipliers. We then check the optimal KKT conditions,

$$\Sigma_{XY} v - 2\Sigma_{XX} \mu u = 0, \quad \Sigma_{XY} u - 2\Sigma_{YY} \sigma v = 0, \quad u^\top \Sigma_{XX} u = 1 \quad \text{and} \quad v^\top \Sigma_{YY} v = 1,$$

which further imply

$$\begin{aligned} & u^\top \Sigma_{XY} v - 2\mu u^\top \Sigma_{XX} u = u^\top \Sigma_{XY} v - 2\mu = 0 \\ \text{and} \quad & v^\top \Sigma_{XY} u - 2\sigma v^\top \Sigma_{YY} v = v^\top \mathbb{E} YX^\top u - 2\sigma = 0. \end{aligned}$$

Solving the above equations, we obtain the optimal Lagrangian multipliers as

$$\mu = \sigma = \frac{1}{2} u^\top \mathbb{E} XY^\top v. \tag{1.7.4}$$

Similarly, we then apply the dual free stochastic gradient method to solve (1.7.2). Specifically, at the $k$-th iteration, we independently sample $(X_k, Y_k)$ and $(\tilde{X}_k, \tilde{Y}_k)$ from $\mathcal{D}$. Then we obtain

$$u_{k+1} = u_k + \eta\Big(\tilde{X}_k\tilde{Y}_k^\top v_k - u_k^\top X_k Y_k^\top v_k \cdot \tilde{X}_k\tilde{X}_k^\top u_k\Big),$$
$$v_{k+1} = v_k + \eta\Big(\tilde{Y}_k\tilde{X}_k^\top u_k - v_k^\top Y_k X_k^\top u_k \cdot \tilde{Y}_k\tilde{Y}_k^\top v_k\Big). \tag{1.7.5}$$

Here we sample two pairs of $X$ and $Y$ to ensure the unbiasedness of the stochastic gradient.

Then we can convert (1.7.5) to ordinary differential equations by taking $\eta \to 0^+$, we get

$$\frac{dU}{dt} = \Sigma_{XY}V - U^\top\Sigma_{XY}V \cdot \Sigma_{XX}U, \quad \frac{dV}{dt} = \Sigma_{XY}^\top U - V^\top\Sigma_{XY}^\top U \cdot \Sigma_{YY}V.$$

Different from PLS, the above ordinary differential equations do not admit a closed form solution, which makes our ODE/SDE-type convergence analysis not applicable in a straightforward manner. A possible alternative approach is to establish the lower bounds for $|\hat{u}^\top U(t)|$ and $|\hat{v}^\top V(t)|$, and further prove that as $t \to \infty$, we have $U(t) \to \hat{u}$ and $V(t) \to \hat{v}$. We will leave this option for further investigation.

Taking our result for PLS as an initial start, we expect more sophisticated and stronger follow-up work that applies to CCA and other online optimization problems with similar structures, which eventually benefits the learning community in both practice and theory.

# CHAPTER 2
# ON LANDSCAPE OF LAGRANGIAN FUNCTIONS AND STOCHASTIC SEARCH FOR CONSTRAINED NONCONVEX OPTIMIZATION

## 2.1 Introduction

We often encounter the following optimization problem in machine learning, signal processing, and stochastic control:

$$\min_{X} f(X) \quad \text{subject to} \quad X \in \Omega, \tag{2.1.1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a loss function, $\Omega :\triangleq \{X \in \mathbb{R}^d : g_i(X) = 0, i = 1, 2, ..., m\}$ denotes a feasible set, $m$ is the number of constraints, and $g_i : \mathbb{R}^d \to \mathbb{R}$'s are the differentiable functions that impose constraints into model parameters. For notational simplicity, we define $\mathcal{G}(X) = [g_1(X), ..., g_m(X)]^\top$ and $\Omega = \{X \in \mathbb{R}^d : \mathcal{G}(X) = 0\}$. Principal component analysis (PCA), canonical correlation analysis (CCA), matrix factorization/sensing/completion, phase retrieval, and many other problems [29, 30, 31, 32, 33, 20, 34] can be viewed as special examples of (2.1.1). Many algorithms have been proposed to solve (2.1.1). For the unconstrained ($\Omega = \mathbb{R}^d$) or a simple constraint $\mathcal{G}(X)$, e.g., the spherical constraint, $\mathcal{G}(X) := \|X\|_2 - 1$, we can apply simple first order algorithms such as the projected gradient descent algorithm [35].

However, when $\mathcal{G}(X)$ is complicated, the aforementioned algorithms are often not applicable or inefficient. This is because the projection to $\Omega$ does not admit a closed form expression and can be computationally expensive in each iteration. To address this issue, we convert (2.1.1) to a min-max problem using the Lagrangian multiplier method. Specifically, instead of solving (2.1.1), we solve the following problem:

$$\min_{X\in\mathbb{R}^d}\max_{Y\in\mathbb{R}^m}\ \mathcal{L}(X,Y) := f(X) + Y^\top \mathcal{G}(X), \tag{2.1.2}$$

where $Y \in \mathbb{R}^m$ is the Lagrangian multiplier. $\mathcal{L}(X,Y)$ is often referred as the Lagrangian function in existing literature [36]. The existing literature on optimization also refers to $X$ as the primal variable and $Y$ as the dual variable. Accordingly, (2.1.1) is called the primal problem. From the perspective of game theory, they can be viewed as two players competing with each other and eventually achieving some equilibrium. When $f(X)$ is convex and $\Omega$ is convex or the boundary of a convex set, the optimization landscape of (2.1.2) is essentially convex-concave, that is, for any fixed $Y$, $\mathcal{L}(X,Y)$ is convex in $X$, and for any fixed $X$, $\mathcal{L}(X,Y)$ is concave in $Y$. Such a landscape further implies that the equilibrium of (2.1.2) is a saddle point, whose primal variable is equivalent to the global optimum of (2.1.1) under strong duality conditions. To solve (2.1.2), we resort to primal-dual algorithms, which iterate over both $X$ and $Y$ (usually in an alternating manner). The global convergence rates to the equilibrium are also established accordingly for these algorithms [37, 38, 39].

When $f(X)$ and $\Omega$ are nonconvex, both (2.1.1) and (2.1.2) become much more computationally challenging, NP-hard in general. Significant progress has been made toward solving the primal problem (2.1.1). For example, [15] show that when certain tensor factorization satisfies the so-called strict saddle properties, one can apply some first order algorithms such as the projected gradient algorithm, and the global convergence in polynomial time can be guaranteed. Their results further motivate many follow-up works, proving that many problems can be formulated as strict saddle optimization problems, including PCA, multiview learning, phase retrieval, matrix factorization/sensing/completion, complete dictionary learning [30, 31, 32, 33, 20, 34]. Note that these strict saddle optimization problems are either unconstrained or just with a simple spherical constraint. However, for many other nonconvex optimization problems, $\Omega$ can be much more complicated. To the best of our knowledge, when $\Omega$ is not only nonconvex but also complicated, the applicable algorithms and convergence guarantees are still largely unknown in existing literature.

To handle the complicated $\Omega$, this chapter proposes to investigate the min-max problem (2.1.2). Specifically, we first define a special class of Lagrangian functions, where the landscape of $\mathcal{L}(X,Y)$ enjoys the following good properties:

- *There exist only two types of equilibria – stable and unstable equilibria. At an unstable equilibrium, $\mathcal{L}(X,Y)$ has negative curvature with respect to the primal variable $X$. More details in Section 2.2.*

- *All stable equilibria correspond to the global optima of the primal problem* (2.1.1).

Both properties are intuitive. On the one hand, the negative curvature in the first property enables the primal variable to escape from the unstable equilibria along some decent direction. On the other hand, the second property ensures that we do not get spurious local optima of (2.1.1), that is all local minima must also be global optima.

We then study a generalized eigenvalue (GEV) problem, which includes CCA, Fisher discriminant analysis (FDA, [40]), sufficient dimension reduction (SDR, [41]) as special examples. Specifically, GEV solves

$$\begin{aligned} X^* = &\underset{X\in\mathbb{R}^{d\times r}}{\operatorname{argmin}}\ f(X) := -\operatorname{tr}(X^\top AX) \\ \text{s.t. }\ X \in &\mathcal{T}_B := \{X\in\mathbb{R}^{d\times r} : X^\top BX = I_r\}, \end{aligned} \tag{2.1.3}$$

where $A, B \in \mathbb{R}^{d\times d}$ are symmetric, $B$ is positive semidefinite. We rewrite (2.1.3) as a min-max problem,

$$\min_X \max_Y \mathcal{L}(X,Y) = -\operatorname{tr}(X^\top AX) + \langle Y, X^\top BX - I_r\rangle, \tag{2.1.4}$$

where $Y \in \mathbb{R}^{r\times r}$ is the Lagrangian multiplier. Theoretically, we show that the Lagrangian function in (2.1.4) exactly belongs to our previously defined class. Motivated by our defined landscape structures, we then solve an online version of (2.1.4), where we can only access independent unbiased stochastic approximations of $A,\ B$ and directly accessing $A$ and $B$ is prohibited. Specifically, at the $k$-th iteration, we only obtain independent $A^{(k)}$ and $B^{(k)}$ satisfying

$$\mathbb{E}A^{(k)} = A \quad \text{and} \quad \mathbb{E}B^{(k)} = B.$$

Computationally, we propose a simple stochastic primal-dual algorithm, which is a stochastic variant of the generalized Hebbian algorithm (GHA, [42]). Theoretically, we establish its asymptotic rate of convergence to stable equilibria for our stochastic GHA (SGHA) based on the diffusion approximations [43]. Specifically, we show that, asymptotically, the solution trajectory of SGHA weakly converges to the solutions of stochastic differential equations (SDEs). By studying the analytical solutions of these SDEs, we further establish the asymptotic sample/iteration complexity of SGHA under certain regularity conditions [43, 27, 20]. To the best of our knowledge, this is the first asymptotic sample/iteration complexity analysis of a stochastic optimization algorithm for solving the online version of GEV problem. Numerical experiments are presented to justify our theory.

Our work is closely related to several recent results on solving GEV problems. For example, [44] propose a multistage semi-stochastic optimization algorithm for solving GEV problems with a finite sum structure. At each optimization stage, their algorithm needs to access the exact $B$ matrix, and compute the approximate inverse of $B$ by solving a quadratic program, which is not allowed in our setting. Similar matrix inversion approaches are also adopted by a few other recently proposed algorithms for solving GEV problem [45, 46]. In contrast, our proposed SGHA is a fully stochastic algorithm, which does not require any matrix inversion.

Moreover, our work is also related to several more complicated min-max problems, such as Markov Decision Process with function approximation, Generative Adversarial Network, multistage stochastic programming and control [47, 48, 49]. Many primal-dual algorithms have been proposed to solve these problems. However, most of these algorithms are even not guaranteed to converge. As mentioned earlier, when the convex-concave structure is missing, the min-max problems go far beyond the existing theories. Moreover, both primal and dual iterations involve sophisticated stochastic approximations (equally or more difficult than our online version of GEV). This chapter makes the attempt on understanding the optimization landscape of these challenging min-max problems. Taking our results as an initial start, we expect more sophisticated and stronger follow-up works that apply to these min-max problems.

**Notations**. Given an integer $d$, we denote $I_d$ as a $d \times d$ identity matrix, $[d] = \{1, 2, \ldots, d\}$. Given an index set $\mathcal{I} \subseteq [d]$ and a matrix $X \in \mathbb{R}^{d\times r}$, we denote $\mathcal{I}^{\perp} = [d] \backslash \mathcal{I}$ as the complement set of $\mathcal{I}$, $X_{:,i}$ ($X_{i,:}$) as the $i$-th column (row) of $X$, $X_{i,j}$ as the $(i, j)$-th entry of $X$, and $X_{:,\mathcal{I}}$ ($X_{\mathcal{I},:}$) as the column (row) submatrix of $X$ indexed by $\mathcal{I}$, $\mathrm{vec}(X) \in \mathbb{R}^{dr}$ as the vectorization of $X$, Col$(X)$ as the column space of $X$, and Null$(X)$ as the null space of $X$. Given a symmetric matrix $X \in \mathbb{R}^{d\times d}$, we denote $\lambda_{\min/\max}(X)$ as its smallest/largest singular value, and denote the eigenvalue decomposition of $X$ as $X = O\Lambda O^{\top}$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots \lambda_d)$ with $\lambda_1 \geq \cdots \geq \lambda_d$, denote $\|X\|_2$ as the spectral norm of $X$. Given two matrices $X$ and $Y$, $X \otimes Y$ as the Kronecker product of $X, Y$.

## 2.2 Characterization of Equilibria

Recall the Lagrangian function in (2.1.2). Then we start with characterizing its equilibria. By KKT conditions, an equilibrium $(X, Y)$ satisfies

$$\nabla_X \mathcal{L}(X, Y) = \nabla_X f(X) + Y^{\top} \nabla_X \mathcal{G}(X) = 0 \quad \text{and} \quad \nabla_Y \mathcal{L}(X, Y) = \mathcal{G}(X) = 0,$$

which only contains the first order information of $\mathcal{L}(X, Y)$. To further distinguish the difference among the equilibria, we define two types of equilibria by the second order information.

**Definition 2.2.1.** Given the Lagrangian function $\mathcal{L}(X,Y)$ in (2.1.2), a point $(X,Y)$ is called:

- **(1)** An ***equilibrium*** of $\mathcal{L}(X,Y)$, if

$$\nabla\mathcal{L}(X,Y) = \begin{bmatrix} \nabla_X\mathcal{L}(X,Y) \\ \nabla_Y\mathcal{L}(X,Y) \end{bmatrix} = 0.$$

- **(2)** An ***equilibrium*** $(X,Y)$ is ***unstable***, if $(X,Y)$ is an equilibrium and $\lambda_{\min}\left(\nabla_X^2\mathcal{L}(X,Y)\right) < 0$.

- **(3)** An ***equilibrium*** $(X,Y)$ is ***stable***, if $(X,Y)$ is an equilibrium, $\nabla_X^2\mathcal{L}(X,Y) \succeq 0$, and $\mathcal{L}(X,Y)$ is strongly convex over a restricted domain.

Note that (2) in Definition 2.2.1 has a similar strict saddle property over a manifold in [15]. The motivation behind Definition 2.2.1 is intuitive. When $\mathcal{L}(X,Y)$ has negative curvature with respect to the primal variable $X$ at an equilibrium, we can find a direction in $X$ to further decrease $\mathcal{L}(X,Y)$. Therefore, a tiny perturbation can break this unstable equilibrium. An illustrative example is presented in Figure 2.1. Moreover, at a stable equilibrium $(X^*,Y^*)$, there is restricted strong convexity, which relates to several conditions, *e.g.*, Polyak Łojasiewicz conditions [50], *i.e.*,

$$\|\nabla_X\mathcal{L}(X,Y^*)\|^2 \geq \mu(\mathcal{L}(X,Y^*) - \mathcal{L}(X^*,Y^*)),$$

for $X$ belonging to a small region near $X^*$ and $\mu > 0$ is a constant, or Error Bound conditions [51]. With this property, we cannot decrease $\mathcal{L}(X,Y)$ along any direction with respect to $X$. Definition 2.2.1 excludes the high order unstable equilibrium, which may exist due to the degeneracy of $\nabla_X^2\mathcal{L}(X,Y)$. Specifically, such a high order unstable equilibrium cannot

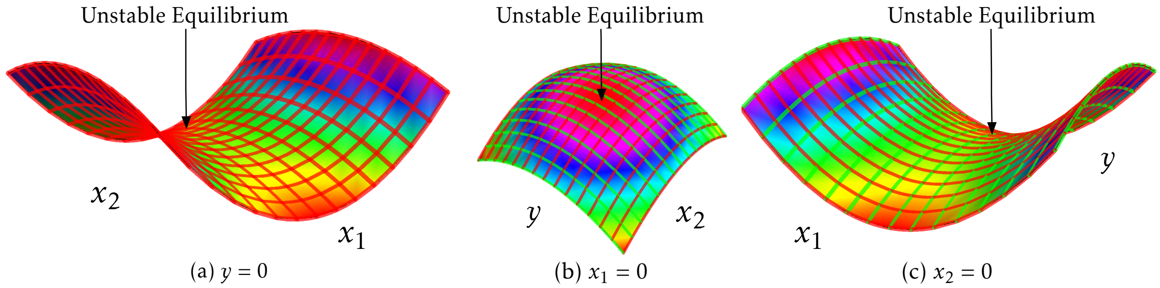



Figure 2.1: An illustration of an unstable equilibrium: $\min_{x_1,x_2} \max_y \mathcal{L}(x_1, x_2, y) = x_1^2 - x_2^2 - y^2$. Notice that $(0, 0, 0)$ is an equilibrium but unstable. For visualization, we show three views: (a) $\mathcal{L}(x_1, x_2, 0)$; (b) $\mathcal{L}(0, x_2, y)$; (c) $\mathcal{L}(x_1, 0, y)$. The red lines correspond to $x_1$ and $x_2$, and the green one corresponds to the $y$.

be identified by the second order information, *e.g.*,

$$\mathcal{L}(x_1, x_2, y) = x_1^3 + x_2^2 + y \cdot (x_1 - x_2).$$

$(0, 0, 0)$ is an equilibrium with a positive semidefinite Hessian matrix. However, it is an unstable equilibria, since a small perturbation to $x_1$ can break this equilibrium. Such an equilibrium makes the landscape highly more complicated. Overall, we consider a specific class of Lagrangian functions throughout the rest of this chapter. They enjoy the following properties:

- All equilibria are either stable or unstable (*i.e.*, no high order unstable equilibria);
- All stable equilibria correspond to the global optima of the primal problem.

As mentioned earlier, the first property ensures that the second order information can identify the type of equilibria. The second property guarantees that we do not get spurious optima for (2.1.1) as long as an algorithm attains a stable equilibrium. Several machine learning problems belong to this class, such as the generalized eigenvalue decomposition problem.

### 2.3 Generalized Eigenvalue Decomposition

We consider the generalized eigenvalue (GEV) problem as a motivating example, which includes CCA, FDA, SDR, etc. as special examples. Recall its min-max formulation (2.1.4):

$$\min_{X\in\mathbb{R}^{d\times r}} \max_{Y\in\mathbb{R}^{r\times r}} \mathcal{L}(X,Y) = -\operatorname{tr}(X^\top AX) + \langle Y, X^\top BX - I_r\rangle.$$

Before we proceed, we impose the following assumption on the problem.

**Assumption 2.3.1.** Given a symmetric matrix $A \in \mathbb{R}^{d\times d}$ and a positive definite matrix $B \in \mathbb{R}^{d\times d}$, the eigenvalues of $\tilde{A} = B^{-\frac{1}{2}}AB^{-\frac{1}{2}}$, denoted by $\lambda_1^{\tilde{A}}, ..., \lambda_d^{\tilde{A}}$, satisfy

$$\lambda_1^{\tilde{A}} \geq \cdots \geq \lambda_r^{\tilde{A}} > \lambda_{r+1}^{\tilde{A}} \geq \cdots \geq \lambda_d^{\tilde{A}}.$$

Such an eigengap assumption avoids the identifiability issue. The full rank assumption on $B$ in Assumption 2.3.1 ensures that the original constrained optimization problem is bounded. This assumption can be further relaxed but require more involved analysis. We will discuss this in Appendix B.1.

To characterize all equilibria of GEV, we leverage the idea of an invariant group. [32] use similar techniques for an unconstrained matrix factorization problem. However, it does not work for the Lagrangian function due to the more complicate landscape. Therefore, we consider a more general invariant group. Moreover, by analyzing the Hessian matrix of $\mathcal{L}(X,Y)$ at the equilibria, we demonstrate that each equilibrium is either unstable or stable and the stable equilibria correspond to the global optima of the primal problem (2.1.3). Therefore, GEV belongs to the class we defined earlier.

### 2.3.1 Invariant Group and Symmetric Property

We first denote the orthogonal group in dimension $r$ as

$$O(r, \mathbb{R}) = \left\{ \Psi \in \mathbb{R}^{r \times r} : \Psi \Psi^\top = \Psi^\top \Psi = I_r \right\}.$$

Notice that for any $\Psi \in O(r, \mathbb{R})$, $\mathcal{L}(X, Y)$ in (2.1.4) has the same landscape with $\mathcal{L}(X\Psi, \Psi^\top Y \Psi)$. This further indicates that given an equilibrium $(X, Y)$, $(X\Psi, \Psi^\top Y \Psi)$ is also an equilibrium. This symmetric property motivates us to characterize the equilibria of $\mathcal{L}(X, Y)$ with an invariant group.

We introduce several important definitions in group theory [52].

**Definition 2.3.2.** Given a group $\mathcal{H}$ and a set $\mathcal{X}$, a map $\phi(\cdot, \cdot)$ from $\mathcal{H} \times \mathcal{X}$ to $\mathcal{X}$ is called the **group action** of $\mathcal{H}$ on $\mathcal{X}$ if $\phi$ satisfies the following two properties:
**Identity:** $\phi(\mathbb{1}, x) = x \ \ \forall x \in \mathcal{X}$, where $\mathbb{1}$ denotes the identity element of $\mathcal{H}$.
**Compatibility:** $\phi(gh, x) = \phi(g, \phi(h, x)) \ \ \forall g, h \in \mathcal{H}, \ \ x \in \mathcal{X}$.

**Definition 2.3.3.** Given a function $f(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, a group $\mathcal{H}$ is a ***stationary invariant group*** of $f$ with respect to two group actions of $\mathcal{H}$, $\phi_1$ on $\mathcal{X}$ and $\phi_2$ on $\mathcal{Y}$, if $\mathcal{H}$ satisfies

$$f(x, y) = f(\phi_1(g, x), \phi_2(g, y)) \ \ \forall x \in \mathcal{X}, \ \ y \in \mathcal{Y}, \ \ \text{and} \ \ g \in \mathcal{H}.$$

For notational simplicity, we denote $\mathcal{G} = O(r, \mathbb{R})$. Given the group $\mathcal{G}$, two sets $\mathbb{R}^{d \times r}$ and $\mathbb{R}^{r \times r}$, we define a group action with $\phi_1$ of $\mathcal{G}$ on $\mathbb{R}^{d \times r}$ and a group action $\phi_2$ of $\mathcal{G}$ on $\mathbb{R}^{r \times r}$ as

$$\phi_1(\Psi, X) = X\Psi \ \ \forall \Psi \in \mathcal{G}, \ X \in \mathbb{R}^{d \times r} \quad \text{and} \quad \phi_2(g, Y) = \Psi^{-1} Y \Psi \ \ \forall \Psi \in \mathcal{G}, \ Y \in \mathbb{R}^{r \times r}.$$

One can check that the orthogonal group $\mathcal{G}$ is a stationary invariant group of $\mathcal{L}(X, Y)$ with respect to two group actions of $\mathcal{G}$, $\phi_1$ on $\mathbb{R}^{d \times r}$ and $\phi_2$ on $\mathbb{R}^{r \times r}$. By this invariant group, we define the equivalence relation between $(X_1, Y_1)$ and $(X_2, Y_2)$, if there exists a $\Psi \in \mathcal{G}$ such that

$$(X_1, Y_1) = (X_2\Psi, \Psi^{-1}Y_2\Psi) = (X_2\Psi, \Psi^\top Y_2\Psi). \tag{2.3.1}$$

To find all equilibria of GEV, we examine the KKT conditions of (2.1.4):

$$2BXY - 2AX = 0 \text{ and } X^\top BX - I_r = 0 \Longrightarrow Y = X^\top AX =: \mathcal{D}(X).$$

Given the eigenvalue decomposition $B = O^B\Lambda^B O^{B\top}$, we denote

$$\tilde{A} = (\Lambda^B)^{-\frac{1}{2}}O^{B\top}AO^B(\Lambda^B)^{-\frac{1}{2}} \quad \text{and} \quad \tilde{X} = (\Lambda^B)^{\frac{1}{2}}O^{B\top}X.$$

We then consider the eigenvalue decomposition $\tilde{A} = O^{\tilde{A}}\Lambda^{\tilde{A}}O^{\tilde{A}\top}$. The following theorem shows the connection between the equilibrium of $\mathcal{L}(X, Y)$ and the column submatrix of $O^{\tilde{A}}$, denoted as $O^{\tilde{A}}_{:,\mathcal{I}}$, where

$$\mathcal{I} \in \mathcal{X}_d^r := \Big\{\{i_1, ..., i_r\} : \{i_1, ..., i_r\} \subseteq [d]\Big\}$$

is the column index set to determine a column submatrix.

**Theorem 2.3.4** (Symmetric Property)**.** Suppose Assumption 2.3.1 holds. Then $(X, \mathcal{D}(X))$ is an equilibrium of $\mathcal{L}(X, Y)$, if and only if $X$ can be written as

$$X = (O^B(\Lambda^B)^{-\frac{1}{2}}O^{\tilde{A}}_{:,\mathcal{I}}) \cdot \Psi,$$

where index $\mathcal{I} \in \mathcal{X}_d^r$ and $\Psi \in \mathcal{G}$.

The proof of Theorem 2.3.4 is provided in Appendix B.1.1. Theorem 2.3.4 implies that there are $\binom{d}{r}$ equilibria of $\mathcal{L}(X, Y)$ under the equivalence relation given in (2.3.1). Each of them corresponds to an $O^{\tilde{A}}_{:,\mathcal{I}}$, where $\mathcal{I} \in \mathcal{X}_d^r$ is the index set. Then whole equilibria set is generated by these $O^{\tilde{A}}_{:,\mathcal{I}}$ with the transformation matrix $O^B(\Lambda^B)^{-\frac{1}{2}}$ and the invariant group action induced by $\mathcal{G}$.

### 2.3.2 Unstable Equilibrium vs. Stable Equilibrium

We further identify the stable and unstable equilibria. Specifically, given $(X, Y)$ as an equilibrium of $\mathcal{L}(X, Y)$, we denote the Hessian matrix of $\mathcal{L}(X, Y)$ with respect to the primal variable $X$ as

$$H_X \triangleq \nabla_X^2 \mathcal{L}(X, Y)|_{Y=\mathcal{D}(X)} \in \mathbb{R}^{dr \times dr}.$$

Then we calculate the eigenvalues of $H_X$. By Definition 2.2.1, $(X, \mathcal{D}(X))$ is unstable if $H_X$ has a negative eigenvalue; Otherwise, we analyze the local landscape at $(X, \mathcal{D}(X))$ to determine whether it is stable or not. The following theorem shows that all equilibria are either stable or unstable and demonstrates how the choice of index set $\mathcal{I}$ corresponds to the unstable and stable equilibria of $\mathcal{L}(X, Y)$.

**Theorem 2.3.5.** Suppose Assumption 2.3.1 holds, and $(X, \mathcal{D}(X))$ is an equilibrium in (2.1.4). By Theorem 2.3.4, $X$ can be represented as $X = (O^B (\Lambda^B)^{-\frac{1}{2}} O^{\tilde{A}}_{:,\mathcal{I}}) \cdot \Psi$ for some $\Psi \in \mathcal{G}$ and $\mathcal{I} \in \mathcal{X}_d^r$.

If $\mathcal{I} \neq [r]$, then $(X, \mathcal{D}(X))$ is an unstable equilibrium with

$$\lambda_{\min}(H_X) \leq \frac{2(\lambda^{\tilde{A}}_{\max \mathcal{I}} - \lambda^{\tilde{A}}_{\min \mathcal{I}^\perp})}{\|X_{:,\min \mathcal{I}^\perp}\|_2^2} < 0,$$

where $\lambda^{\tilde{A}}_{\max \mathcal{I}} = \max_{i \in \mathcal{I}} \lambda^{\tilde{A}}_i$, and $\lambda^{\tilde{A}}_{\min \mathcal{I}} = \min_{i \in \mathcal{I}} \lambda^{\tilde{A}}_i$, $\lambda^{\tilde{A}}_i$ is the $i$-th leading eigenvalue of $\tilde{A}$.

Otherwise, we have $H_X \succeq 0$ and $\operatorname{rank}(H_X) = d \times r - r(r-1)/2$. Moreover, $(X, \mathcal{D}(X))$ is a stable equilibrium of min-max problem (2.1.4).

The proof of Theorem 2.3.5 is provided in Appendix B.1.2. Theorem 2.3.5 indicates that when $\tilde{X} = O^{\tilde{A}}_{:,[r]}$, that is, the eigenvectors of $\tilde{A}$ corresponding to the $r$ largest eigenvalues, $(X, \mathcal{D}(X))$ is a stable equilibrium of $\mathcal{L}(X, Y)$, where $X = (O^B (\Lambda^B)^{-\frac{1}{2}} O^{\tilde{A}}_{:,\mathcal{I}})) \cdot$

$\Psi$ for some $\Psi \in \mathcal{G}$. Although $H_X$ is degenerate at this equilibrium, all directions in $\text{Null}(H_X)$ essentially point to the primal variables of other stable equilibria. Excluding these directions, the rest all have positive curvature, which implies that this equilibrium is stable. Moreover, such an $X$ corresponds to the optima of (2.1.3). When $\mathcal{I} \neq [r]$, due to the negative curvature, these equilibria are unstable. Therefore, all stable equilibria of $\mathcal{L}(X, Y)$ correspond to the global optima in (2.1.3) and other equilibria are unstable, which further indicates that GEV belongs to the class we defined earlier.

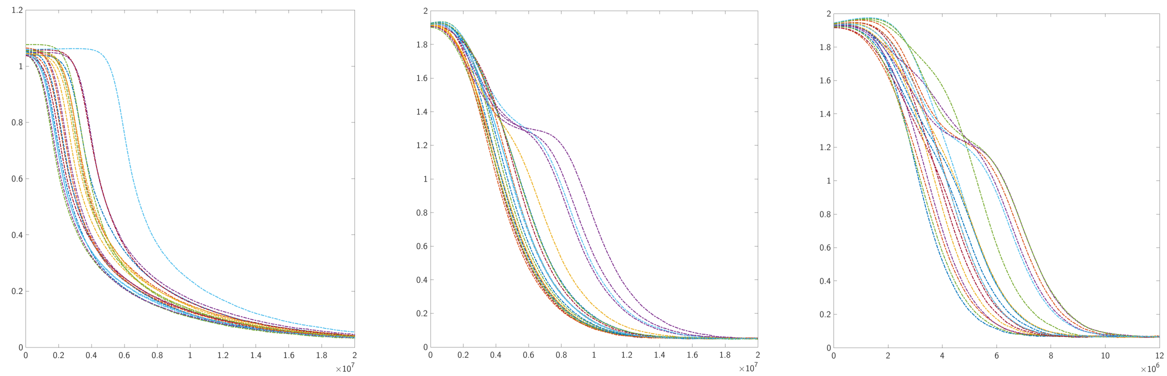
## 2.4 Stochastic Search for Online GEV

For GEV, we propose a fully stochastic primal-dual algorithm to solve (2.1.4), which only requires access to the stochastic approximations of $A$ and $B$ matrices. This is very different from other existing semi-stochastic algorithms that require to access the exact $B$ matrix [44]. Specifically, we propose a stochastic variant of the generalized Hebbian algorithm (GHA), also referred as Sanger's rule in existing literature [19], to solve (2.1.4). For online setting, accessing the exact $A$ and $B$ is prohibitive and we only get $A^{(k)} \in \mathbb{R}^{d\times d}$ and $B^{(k)} \in \mathbb{R}^{d\times d}$ that are independently sampled from the distribution associated with $A$ and $B$ at the $k$-th iteration. Our proposed SGHA updates primal and dual variables as follows:

$$\text{Primal Update: } X^{(k+1)} \leftarrow X^{(k)} - \eta \cdot \underbrace{\left(B^{(k)}X^{(k)}Y^{(k)} - A^{(k)}X^{(k)}\right)}_{\text{Stochastic Approximation of } \nabla_X \mathcal{L}(X^{(k)}, Y^{(k)})}, \tag{2.4.1}$$

$$\text{Dual Update: } Y^{(k+1)} \leftarrow \underbrace{X^{(k)\top}A^{(k)}X^{(k)}}_{\text{Stochastic Approximation of } X^{(k)\top}AX^{(k)}}, \tag{2.4.2}$$

where $\eta > 0$ is a step size parameter. Note that the primal update is a stochastic gradient descent step, while the dual update is motivated by the KKT conditions of (2.1.4). SGHA is simple and easy to implement. The constraint is naturally handled by the dual update. Further, motivated by the the landscape of GEV, we analyze the algorithm by diffusion approximations and obtain the asymptotical sample complexity.



(a) Setting (1). (b) Setting (2). (c) Setting (3).

Figure 2.2: Plots of the optimization error $\|B^{1/2}X^{(t)}X^{(t)\top}B^{1/2} - B^{1/2}X^*X^{*\top}B^{1/2}\|_{\mathrm{F}}$ over SGHA iterations on synthetic data of 20 random data generations under different settings of parameters.

### 2.4.1 Numerical Evaluations

We first provide numerical evaluations to illustrate the effectiveness of SGHA, and then provide an asymptotic convergence analysis of SGHA. We choose $d = 500$ and select three different settings:

- **Setting(1)** : $\eta = 10^{-4}$, $r = 1$, $A_{ii} = 1/100 \ \ \forall i \in [d]$, $A_{ij} = 0.5/10$ and $B_{ij} = 0.5^{|i-j|}/3 \ \ \forall i \neq j$;

- **Setting(2)** : $\eta = 5 \times 10^{-5}$, $r = 3$, and randomly generate an orthogonal matrix $U \in \mathbb{R}^{d \times d}$ such that $A = U \cdot \mathrm{diag}(1, 1, 1, 0.1, ..., 0.1) \cdot U^\top$ and $B = U \cdot \mathrm{diag}(2, 2, 2, 1, ..., 1) \cdot U^\top$;

- **Setting(3)** : $\eta = 2.5 \times 10^{-5}$, $r = 3$, and randomly generate two orthogonal matrices $U, V \in \mathbb{R}^{d \times d}$ such that $A = U \cdot \mathrm{diag}(1, 1, 1, 0.1, ..., 0.1) \cdot U^\top$ and $B = V \cdot \mathrm{diag}(2, 2, 2, 1, ..., 1) \cdot V^\top$.

At the $k$-th iteration of SGHA, we independently sample $40$ random vectors from $N(0, A)$ and $N(0, B)$ respectively. Accordingly, we compute the sample covariance matrices $A^{(k)}$ and $B^{(k)}$ as the approximations of $A$ and $B$. We repeat numerical simulations under each setting for 20 times using random data generations, and present all results in Figure 2.2. The horizontal axis corresponds to the number of iterations, and the vertical axis corresponds to the optimization error

$$\|B^{1/2}X^{(t)}X^{(t)\top}B^{1/2} - B^{1/2}X^{*}X^{*\top}B^{1/2}\|_{\mathrm{F}}.$$

Our experiments indicate that SGHA converges to a global optimum in all settings.

### 2.4.2 Convergence Analysis for Commutative $A$ and $B$

As a special case, we first prove the convergence of SGHA for GEV with $r = 1$, and $A$ and $B$ are commutative. We will discuss more on noncommutative cases and $r > 1$ in the next section. Before we proceed, we introduce our assumptions on the problem.

**Assumption 2.4.1.** We assume that the following conditions hold:

- **(a):** $A^{(k)}$'s and $B^{(k)}$'s are independently sampled from two different distributions $\mathcal{D}_A$ and $\mathcal{D}_B$ respectively, where $\mathbb{E}A^{(k)} = A$ and $\mathbb{E}B^{(k)} = B \succ 0$;

- **(b):** $A$ and $B$ are commutative, *i.e.*, there exists an orthogonal matrix $O$ such that $A = O\Lambda^{A}O^{\top}$ and $B = O\Lambda^{B}O^{\top}$, where $\Lambda^{A} = \mathrm{diag}(\lambda_1, ..., \lambda_d)$ and $\Lambda^{B} = \mathrm{diag}(\mu_1, ..., \mu_d)$ are diagonal matrices with $\lambda_j \neq 0$;

- **(c):** $A^{(k)}$ and $B^{(k)}$ satisfy the moment conditions, that is, for some generic constants $C_0$ and $C_1$, $\mathbb{E}\|A^{(k)}\|_2^2 \leq C_0$ and $\mathbb{E}\|B^{(k)}\|_2^2 \leq C_1$.

Note that (a) and (c) in Assumption 2.4.1 are mild, but (b) is stringent. For convenience of analysis, we combine (2.4.1) and (2.4.2) as

$$X^{(k+1)} \leftarrow X^{(k)} - \eta\big(B^{(k)}X^{(k)}X^{(k)\top} - I_d\big)A^{(k)}X^{(k)}. \tag{2.4.3}$$

We remark that (2.4.3) is very different from existing optimization algorithms over the generalized Stiefel manifold. Specifically, computing the gradient over the generalized Stiefel manifold requires $B^{-1}$, which is not allowed in our setting. For notational convenience, we further denote

$$\Lambda = (\Lambda^B)^{-\frac{1}{2}}\Lambda^A(\Lambda^B)^{-\frac{1}{2}} = \mathrm{diag}\left(\frac{\lambda_1}{\mu_1}, ..., \frac{\lambda_d}{\mu_d}\right) =: \mathrm{diag}(\beta_1, \cdots, \beta_d).$$

Without loss of generality, we assume $\beta_1 > \beta_2 \geq \beta_3 \geq \cdots \geq \beta_d$, and $\beta_i \neq 0 \;\; \forall i \in [d]$. Note that $\mu_i$ and $\lambda_i$, however, are not necessarily to be monotonic. We denote

$$\mu_{\min} = \min_{i\neq 1} \mu_i, \quad \mu_{\max} = \max_{i \neq 1} \mu_i, \quad \text{and} \quad \mathsf{gap} = \beta_1 - \beta_2.$$

Denote $W^{(k)} = (\Lambda^B)^{\frac{1}{2}} O X^{(k)}$. One can verify that (2.4.3) can be rewritten as follows:

$$W^{(k+1)} \leftarrow W^{(k)} - \eta\Big((\Lambda^B)^{\frac{1}{2}}\hat{\Lambda}_B^{(k)}(\Lambda^B)^{-\frac{1}{2}} \cdot W^{(k)}W^{(k)\top} - \Lambda^B\Big) \cdot \tilde{\Lambda}^{(k)}W^{(k)}, \tag{2.4.4}$$

where $\hat{\Lambda}_B^{(k)} = O^\top B^{(k)} O$ and $\tilde{\Lambda}^{(k)} = O^\top B^{-\frac{1}{2}} A^{(k)} B^{-\frac{1}{2}} O$. Note that $W^* = (1, \underbrace{0, 0, ..., 0}_{(d-1)})^\top$ corresponds to the optimal solution of (2.1.3).

By diffusion approximation, we show that our algorithm converges through three Phases:

- **Phase I:** Given an initial near a saddle point, we show that after rescaling of time properly, the algorithm can be characterized by a stochastic differential equation (SDE). Such an SDE further implies our algorithm can escape from the saddle fast;

- **Phase II:** We show that away from the saddle, the trajectory of our algorithm can be approximated by an ordinary differential equation (ODE);

- **Phase III:** We first show that after Phase II, the norm of solution converges to a constant. Then, the algorithm can be characterized by an SDE, like Phase I. By the SDE, we analyze the error fluctuation when the solution is within a small neighborhood of the global optimum.

Overall, we obtain an asymptotic sample complexity.

**ODE Characterization:** To demonstrate an ODE characterization for the trajectory of our algorithm, we introduce a continuous time random process

$$w^{(\eta)}(t) := W^{(k)},$$

where $k = \lfloor \frac{t}{\eta} \rfloor$ and $\eta$ is the step size in (2.4.3). For notational simplicity, we drop $(t)$ when it is clear from the context. Instead of showing a global convergence of $w^{(\eta)}$, we show that the quantity

$$v_{i,j}^{(\eta)} = \frac{(w_i^{(\eta)})^{\mu_j}}{(w_j^{(\eta)})^{\mu_i}}$$

converges to an exponential decay function, where $v_i^{(\eta)}$ is the $i$-th component (coordinate) of $w^{(\eta)}$.

**Lemma 2.4.2.** Suppose that Assumption 2.4.1 holds and the initial solution is away from any saddle point, *i.e.*, given pre-specified constants, $\tau > 0$ and $\delta < \frac{1}{2}$, there exist $i, j$ such that

$$i \neq j, \quad |w_j^{(\eta)}| > \tau, \quad \text{and} \quad |w_i^{(\eta)}| > \eta^{\frac{1}{2}+\delta}.$$

As $\eta \to 0$, $v_{k,j}^{(\eta)}$ weakly converges to the solution of the following ODE:

$$dx_{k,j} = x_{k,j} \cdot (\mu_j \mu_k (\beta_k - \beta_j))\, dt \ \ \forall k \neq j. \tag{2.4.5}$$

The proof of Lemma 2.4.2 is provided in Appendix B.2.1. Lemma 2.4.2 essentially implies the global convergence of SGHA. Specifically, the solution of (2.4.5) is

$$x_{k,j}(t) = x_{k,j}(0) \cdot \exp\left(\mu_j \mu_k \left(\beta_k - \beta_j\right) t\right) \ \ \forall k \neq j,$$

where $x_{k,j}(0)$ is the initial value of $v_{k,j}^{(\eta)}$. In particular, we consider $j = 1$. Then, as $t \to \infty$, the dominating component of $w$ will be $w_1$.

The ODE approximation of the algorithm implies that after long enough time, *i.e.*, $t$ is large enough, the solution of the algorithm can be arbitrarily close to a global optimum. Nevertheless, to obtain the asymptotic "convergence rate", we need to study the variance of the trajectory at time $t$. Thus, we resort to the following SDE-based approach for a more precise characterization.

**SDE Characterization:** We notice that such a variance with order $\mathcal{O}(\eta)$ vanishes as $\eta \to 0$. To characterize this variance, we rescale the updates by a factor of $\eta^{-\frac{1}{2}}$, *i.e.*, by defining a new process as $z^{(\eta)} = \eta^{-\frac{1}{2}} w^{(\eta)}$. After rescaling, the variance of $z^{(\eta)}$ is of order $\mathcal{O}(1)$. The following lemma characterizes how the algorithm escapes from the saddle, *i.e.*, $w^{(\eta)}(0) \approx e_i$, where $i \neq 1$, in Phase I.

**Lemma 2.4.3.** Suppose Assumption 2.4.1 holds and the initial is close to a saddle point , *i.e.*, $z_j^{(\eta)}(0) \approx \eta^{-\frac{1}{2}}$ and $z_i^{(\eta)}(0) \approx 0$ for $i \neq j$. Then for any $C > 0$, there exist $\tau > 0$ and $\eta' > 0$ such that

$$\sup_{\eta < \eta'} \mathbb{P}(\sup_t |z_i^{(\eta)}(t)| \leq C) \leq 1 - \tau. \tag{2.4.6}$$

Here we provide the proof sketch and leave the whole proof of Lemma 2.4.3 in Appendix B.2.2.

*Proof Sketch.* We prove this argument by contradiction. Assume the conclusion does not hold, that is there exists a constant $C > 0$, such that for any $\eta' > 0$ we have

$$\sup_{\eta \leq \eta'} \mathbb{P}(\sup_t |z_i^{(\eta)}(t)| \leq C) = 1.$$

That implies there exists a sequence $\{\eta_n\}_{n=1}^{\infty}$ converging to $0$ such that

$$\lim_{n \to \infty} \mathbb{P}(\sup_t |z_i^{(\eta_n)}(t)| \leq C) = 1. \tag{2.4.7}$$

Then we show $\{z_i^{(\eta_n)}(\cdot)\}_n$ is tight and thus converges weakly. Furthermore, $\{z_i^{(\eta_n)}(\cdot)\}_n$

weakly converges to a stochastic differential equation,

$$dz_j(t) = (-\beta_j \mu_i \cdot z_i + \lambda_i z_i)\, dt + \sqrt{G_{j,i}} dB(t) \ \text{ for } j \in [d] \backslash \{i\}, \tag{2.4.8}$$

where $G_{j,i} = \mathbb{E}\Big(\big(\hat{\Lambda}_B^{(k)}\big)_{j,i} \cdot \sqrt{\mu_j/\mu_i} \cdot \tilde{\Lambda}_{i,i} - \mu_j \tilde{\Lambda}_{j,i}\Big)^2$ and $B(t)$ is a standard Brownian motion. We compute the solution of this stochastic differential equation and then show (2.4.6) holds. □

Note that (2.4.8) is a Fokker-Plank equation, whose solution is an Ornstein-Uhlenbeck (O-U) process [53] as follows:

$$z_j(t) = \Big[\underbrace{z_j(0) + \sqrt{G_{j,i}} \int_0^t \exp\left[\mu_j\left(\beta_i - \beta_j\right) s\right] dB(s)}_{Q_1}\Big] \cdot \exp\left[-\mu_j\left(\beta_i - \beta_j\right) t\right]. \tag{2.4.9}$$

We consider $j = 1$. Note that $Q_1$ is essentially a random variable with mean $z_j(0)$ and variance smaller than $\frac{G_{1,i}\mu_1}{2(\beta_1 - \beta_i)}$. However, the larger $t$ is, the closer its variance gets to this upper bound. Moreover, the term $\exp\big[\mu_1(\beta_1 - \beta_i)t\big]$ essentially amplifies $Q_1$ by a factor exponentially increasing in $t$. This tremendous amplification forces $z_1(t)$ to quickly get away from $0$, as $t$ increases, which indicates that the algorithm will escape from the saddle. Further, the following lemma characterizes the local behavior of the algorithm near the optimal.

**Lemma 2.4.4.** Suppose that Assumption 2.4.1 holds and the initial solution is close to an optimal solution, that is, given pre-specified constants $\kappa$ and $\delta < \frac{1}{2}$, we have $\frac{|w_1^{(\eta)}|^2}{\|w^{(\eta)}\|_2^2} > 1 - \kappa\eta^{1+2\delta}$. As $\eta \to 0$, then we have $\|w^{(\eta)}(t)\|_2 \xrightarrow{t \to \infty} 1$ and $z_i^{(\eta)}$ weakly converges to the solution of the following SDE:

$$dz_i(t) = (-\beta_1 \cdot \mu_i z_i + \lambda_i z_i)\, dt + \sqrt{G_{i,1}} dB(t) \ \text{ for } i \neq 1, \tag{2.4.10}$$

where $G_{i,1} = \mathbb{E}\big((\hat{\Lambda}_B)_{i,1} \cdot \sqrt{\mu_i/\mu_1} \cdot \tilde{\Lambda}_{1,1} - \mu_i \Lambda_{i,1}\big)^2$, and $B(t)$ is a standard Brownian motion.

The proof of Lemma 2.4.4 is provided in Appendix B.2.3. The solution of (2.4.10) is as follows:

$$z_i(t) = \sqrt{G_{i,1}} \int_0^t \exp\left[\mu_i \left(\beta_1 - \beta_i\right)(s - t)\right] dB(s) \\ + z_i(0) \cdot \exp\left[-\mu_i \left(\beta_1 - \beta_i\right) t\right]. \tag{2.4.11}$$

Note the second term of the right hand side in (2.4.11) decays to 0, as time $t \to \infty$. The rest is a pure random walk. Thus, the fluctuation of $z_i(t)$ is essentially the error fluctuation of the algorithm after sufficiently long time.

Combining Lemma 2.4.2, 2.4.3, and 2.4.4, we obtain the following theorem.

**Theorem 2.4.5.** Suppose Assumption 2.4.1 holds. Given a sufficiently small error $\epsilon > 0$, $\phi = \sum_{i=1}^{d} G_{i,1}$, and

$$\eta \asymp \frac{\epsilon \cdot \mu_{\min} \cdot \mathsf{gap}}{\phi},$$

we need

$$T \asymp \frac{\mu_{\max}/\mu_{\min}}{\mu_1 \cdot \mathsf{gap}} \log\left(\eta^{-1}\right) \tag{2.4.12}$$

such that with probability at least $\frac{5}{8}$, $\|w(T) - W^*\|_2^2 \leq \epsilon$, where $W^*$ is the optima of (2.1.3).

The proof of Theorem 2.4.5 is provided in Appendix B.2.4. Theorem 2.4.5 implies that asymptotically, our algorithm yields an iterations of complexity:

$$N \asymp \frac{T}{\eta} \asymp \frac{\phi \cdot \mu_{\max}/\mu_{\min}}{\epsilon \cdot \mu_1 \cdot \mu_{\min} \cdot \mathsf{gap}^2} \log\left(\frac{\phi}{\epsilon \cdot \mu_{\min} \cdot \mathsf{gap}}\right),$$

which not only depends on the gap, *i.e.*, $\beta_1 - \beta_2$, but also depends on $\frac{\mu_{\max}}{\mu_{\min}}$, which is the condition number of $B$ in the worst case. As can be seen, for an ill-conditioned $B$, the problem (2.1.3) is more difficult to solve.

### 2.4.3 When $A$ and $B$ are Noncommutative?

Unfortunately, when $A$ and $B$ are noncommutative, the analysis is more difficult, even for $r = 1$. Recall that the optimization landscape of the Lagrangian function in (2.1.4) enjoys a nice geometric property: At an unstable equilibrium, the negative curvature with respect to the primal variable encourages the algorithm to escape. Specifically, suppose the algorithm is initialized at an unstable equilibrium $(X^{(0)}, Y^{(0)})$, the descent direction for $X^{(0)}$ is determined by the eigenvectors of

$$H_{X^{(0)}} = A + Y^{(0)}B$$

associated with the negative eigenvalues. After one iteration, we obtain $(X^{(1)}, Y^{(1)})$. The Hessian matrix becomes

$$H_{X^{(1)}} = A + Y^{(1)}B.$$

Since $Y^{(1)} = X^{(0)\top}A^{(0)}X^{(0)}$ is a stochastic approximation, the random noise can make $Y^{(1)}$ significantly different from $Y^{(0)}$. Thus, the eigenvectors of $H_{X^{(1)}}$ associated with the negative eigenvalues can be also very different from those of $H_{X^{(0)}}$. This phenomenon can seriously confuse the algorithm about the descent direction of the primal variable. We remark that such an issue does not appear if we assume $A$ and $B$ are commutative. We suspect that this is very likely an artifact of our proof technique, since our numerical experiments have provided some empirical evidences of the convergence of SGHA.

## 2.5 Discussions

Here we briefly discuss a few related works:

- [32] propose a framework for characterizing the stationary points in the unconstrained nonconvex matrix factorization problem, while our studied generalized eigenvalue problem is constrained. Different from their analysis, we analyze the optimization landscape of the corresponding Lagrangian function. When characterize the stationary points, we need to take both primal and dual variables into consideration, which is technically more challenging.

- [44] also consider the (off-line) generalized eigenvalue problem but in a finite sum form. Unlike our studied online setting, they access exact $A$ and $B$ in each iteration. Specifically, they need to access exact $A$ and $B$ to compute an approximate inverse of $B$ to find the descent direction. Meanwhile, they also need a modified Gram Schmidt process, which also requires accessing exact $B$, to maintain the solution on the generalized Stiefel manifold (defined by $X^\top BX = I_r$ via exact $B$, [54]). Our proposed stochastic search, however, is a full stochastic primal-dual algorithm, which neither require accessing exact $A$ and $B$, nor enforcing the the primal variables to stay on the manifold.

# CHAPTER 3
# A HIERARCHICAL EXPECTED IMPROVEMENT METHOD FOR BAYESIAN OPTIMIZATION

## 3.1 Introduction

Bayesian optimization (BO) is a widely-used optimization framework, which has broad applicability in a variety of problems, including rocket engine design [55], nanowire yield optimization [56], and surgery planning [57]. BO aims to solve the following black-box optimization problem:

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}\in\Omega} f(\mathbf{x}). \tag{3.1.1}$$

Here, $\mathbf{x} \in \mathbb{R}^d$ are the input variables, and $\Omega \subset \mathbb{R}^d$ is the feasible domain for optimization. The key challenge in (3.1.1) is that the objective function $f(\cdot) : \Omega \to \mathbb{R}$ is assumed to be black-box: it admits no closed-form expression, and evaluations of $f$ typically require expensive simulations or experiments. For such problems, an optimization procedure should find a good solution to (3.1.1) given limited function evaluations. BO achieves this by first assigning to $f$ a *prior model* capturing prior beliefs on the objective function, then sequentially querying $f$ at points which maximize the *acquisition function* – the posterior expected utility of a new point. This provides a principled way for performing the so-called *exploration-exploitation trade-off* [58]: exploring the black-box function over $\Omega$, and exploiting the fitted function when appropriate for optimization.

Much of the literature on BO can be categorized by (i) the prior stochastic model assumed on $f$, and (ii) the utility function used for sequential sampling. For (i), the most popular stochastic model is the Gaussian process (GP) model [59]. Under a GP model, several well-known BO methods have been derived using different utility functions for (ii). These include the expected improvement (EI) method [60, 61], the upper confidence bound (UCB) method [62], and the Knowledge Gradient method [63, 64]. Of these, EI is arguably the most popular method, since it admits a simple *closed-form* acquisition function, which can be efficiently optimized to yield subsequent query points on $f$. EI has been subsequently developed for a variety of black-box optimization problems, including multi-fidelity optimization [65], constrained optimization [66], and parallel/batch-sequential optimization [67].

Despite the popularity of EI, it does have key limitations. One such limitation is that it is too *greedy* [68]: EI focuses nearly all sampling efforts near the optima of the *fitted* GP model, and does not sufficiently explore other regions. In terms of the exploration-exploitation trade-off [58], EI over-exploits the fitted model on $f$, and under-explores the feasible domain; this causes the procedure to get stuck in local optima and fail to converge to a global optimum $\mathbf{x}^*$ [69]. There have some efforts on remedying this greediness. [70] proposed a fully Bayesian EI, where GP model parameters are sampled using Markov chain Monte Carlo (MCMC); this incorporates parameter uncertainty within EI and encourages exploration. [71] proposed a variation of EI under an additive Bayesian model, which encourages exploration by increasing model uncertainty. Such methods, however, require expensive MCMC sampling, which can be very computationally expensive to integrate for maximizing the acquisition function. This computational burden diminishes a key advantage of EI: efficient queries via a closed-form criterion.

Another remedy, proposed by [69], is to artificially inflate the maximum-likelihood estimator of the GP variance, and use this inflated estimate within the EI acquisition function for sequential sampling. This, in effect, encourages exploration of the black-box function by inflating the uncertainty of the fitted model. The work further proposes an "$\epsilon$-greedy" modification of the EI, where at each sampling iteration, one selects the next point uniformly with probability $\epsilon \in (0, 1)$. With a sufficiently large $\epsilon$, this again forces the procedure to explore the feasible domain. However, even with such modifications, this approach may perform poorly for expensive black-box problems with very limited samples. One reason is because these modifications, while encouraging further exploration, does so in a rather *ad-hoc* manner; when evaluations are limited, randomly placed samples can be sub-optimal for exploration of the optimization space, particularly in high dimensions. This is demonstrated later in a suite of simulation experiments.

To address this, we propose a novel hierarchical EI (HEI) framework, which corrects the greediness of EI while preserving a closed-form acquisition function. The key idea is the use of a *hierarchical* GP model for $f$ [72], which assigns hierarchical priors on process parameters. Under this, we show that HEI has a closed-form acquisition function, which provides a principled approach for encouraging exploration via hierarchical Bayesian modeling. With this closed-form expression, we introduce hyperparameter estimation methods which allow HEI to mimic a fully Bayesian optimization procedure, while avoiding expensive MCMC sampling steps. We then prove that, under certain prior specifications, HEI converges to a global optimum $\mathbf{x}^*$ over a broad function space for $f$. This addresses the over-greediness of EI, which can fail to find any global optimum even for smooth $f$. We further prove that HEI achieves a near-minimax convergence rate, under regularity assumptions which can be easily checked.

Finally, we note that a special case of HEI, called the Student EI (SEI), was proposed earlier in [73]. The proposed HEI has several key advantages over the SEI: the HEI incorporates uncertainty on process nonstationarity, and can mimic a fully Bayesian optimization procedure via hyperparameter estimation. We also show that the HEI has provable global convergence and convergence rates for optimization, whereas the SEI (with the recommended hyperparameter specification) can fail to converge to a global optimum $\mathbf{x}^*$. Numerical experiments and a semiconductor manufacturing application show the improved performance of the HEI over existing BO methods, including the aforementioned SEI and $\epsilon$-greedy approaches.

The chapter is organized as follows: Section 3.2 reviews the GP model and the EI method. Section 3.3 presents the HEI method and contrasts it with existing methods. Section 3.4 provides methodological developments on hyperparameter specification and basis selection. Section 3.5 proves the global convergence for HEI and its associated convergence rates. Sections 3.6 and 3.7 compare HEI with existing methods in a suite of numerical experiments and for a semiconductor manufacturing problem, respectively. Concluding remarks are given in Section 3.8.

## 3.2 Background and Motivation

We first introduce the GP model, then review the EI method and its deficiencies, which will help motivate the proposed HEI method.

### 3.2.1 Gaussian Process Modeling

We first model the black-box objective function $f$ as the following Gaussian process model:

$$f(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x}), \quad \mu(\mathbf{x}) = \mathbf{p}^\top(\mathbf{x})\boldsymbol{\beta}, \quad Z(\mathbf{x}) \sim \mathrm{GP}(0, \sigma^2 K), \tag{3.2.1}$$

where $\mu(\mathbf{x})$ is the mean function of the process, $\mathbf{p}(\mathbf{x}) = [p_1(\mathbf{x}), \cdots, p_q(\mathbf{x})]^\top$ are the $q$ basis functions for $\mu(\mathbf{x})$, and $\boldsymbol{\beta} \in \mathbb{R}^q$ are its corresponding coefficients. Here, we assume the residual term $Z(\mathbf{x})$ follows a stationary Gaussian process prior [59] with mean zero, process variance $\sigma^2$ and correlation function $K(\cdot,\cdot)$, which we denote as $Z(\mathbf{x}) \sim \mathrm{GP}(0, \sigma^2 K)$. The model (3.2.1) is known as the *universal kriging* (UK) model in the geostatistics literature [74]. When there is no trend, i.e., $p(\mathbf{x}) = 1$, this model reduces to the so-called *ordinary kriging* (OK) model [75].

Suppose function values $y_i = f(\mathbf{x}_i)$ are observed at inputs $\mathbf{x}_i$, yielding data $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$. Let $\mathbf{y}_n = (y_i)_{i=1}^n$ be the vector of observed function values, $\mathbf{k}_n(\mathbf{x}) = (K(\mathbf{x}, \mathbf{x}_i))_{i=1}^n$ be the correlation vector between the unobserved response $f(\mathbf{x})$ and observed responses $\mathbf{y}_n$, $\mathbf{K}_n = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^n$ be the correlation matrix for observed points, and $\mathbf{P}_n = [\mathbf{p}(\mathbf{x}_1), \cdots, \mathbf{p}(\mathbf{x}_n)]^\top$ be the model matrix for observed points. Then the posterior distribution of $f(\mathbf{x})$ at an unobserved input $\mathbf{x}$ has the closed form expression [59]:

$$[f(\mathbf{x})\big|\mathcal{D}_n] \sim \mathcal{N}\Big(\hat{f}_n(\mathbf{x}), \sigma^2 s_n^2(\mathbf{x})\Big) \ . \tag{3.2.2}$$

Here, the posterior mean $\hat{f}_n(\mathbf{x})$ is given by:

$$\hat{f}_n(\mathbf{x}) = \mathbf{p}^\top(\mathbf{x})\hat{\boldsymbol{\beta}}_n + \mathbf{k}_n^\top(\mathbf{x})\mathbf{K}_n^{-1}\left(\mathbf{y}_n - \mathbf{P}_n\hat{\boldsymbol{\beta}}_n\right), \tag{3.2.3}$$

the posterior variance $\sigma^2 s_n^2(\mathbf{x})$ is given by:

$$\sigma^2 s_n^2(\mathbf{x}) = \sigma^2\left(K(\mathbf{x},\mathbf{x}) - \mathbf{k}_n^\top(\mathbf{x})\mathbf{K}_n^{-1}\mathbf{k}_n(\mathbf{x}) + \mathbf{h}_n^\top(\mathbf{x})\mathbf{G}_n^{-1}\mathbf{h}_n(\mathbf{x})\right), \tag{3.2.4}$$

where $\hat{\boldsymbol{\beta}}_n = \mathbf{G}_n^{-1}\mathbf{P}_n^\top\mathbf{K}_n^{-1}\mathbf{y}_n$ is the maximum likelihood estimator (MLE) for $\boldsymbol{\beta}$, $\mathbf{G}_n = \mathbf{P}_n^\top\mathbf{K}_n^{-1}\mathbf{P}_n$ and $\mathbf{h}_n(\mathbf{x}) = \mathbf{p}(\mathbf{x}) - \mathbf{P}_n^\top\mathbf{K}_n^{-1}\mathbf{k}_n(\mathbf{x})$. These expressions can be equivalently viewed as the best linear unbiased estimator of $f(\mathbf{x})$ and its variance [59].

### 3.2.2 Expected Improvement

The EI method [61] then makes use of the above closed-form equations to derive a closed-form acquisition function. Let $y_n^* = \min_{i=1}^n y_i$ be the current best objective value, and let $(y_n^* - f(\mathbf{x}))_+ = \max\{y_n^* - f(\mathbf{x}), 0\}$ be the *improvement* utility function. Given data $\mathcal{D}_n$, the expected improvement acquisition function can be written as:

$$\mathbb{E}_{f|\mathcal{D}_n}(y_n^* - f(\mathbf{x}))_+ = I_n(\mathbf{x})\Phi\left(\frac{I_n(\mathbf{x})}{\sigma s_n(\mathbf{x})}\right) + \sigma s_n(\mathbf{x})\phi\left(\frac{I_n(\mathbf{x})}{\sigma s_n(\mathbf{x})}\right). \tag{3.2.5}$$

Here, $\phi(\cdot)$ and $\Phi(\cdot)$ denote the probability density function (p.d.f.) and cumulative density function (c.d.f.) of the standard normal distribution, respectively, and $I_n(\mathbf{x}) = y_n^* - \hat{f}_n(\mathbf{x})$. For an unobserved point $\mathbf{x}$, $\mathbb{E}_{f|\mathcal{D}_n}(y_n^* - f(\mathbf{x}))_+$ can be interpreted as the expected improvement to the current best objective value, if the next query is at point $\mathbf{x}$.

In order to compute the acquisition function in Equation (3.2.5), we would need to know the process variance $\sigma^2$. In practice, however, this is typically unknown and needs to be estimated from data. A standard approach [69] is to compute its MLE:

$$\hat{\sigma}_n^2 = \frac{1}{n}(\mathbf{y}_n - \mathbf{P}_n\hat{\boldsymbol{\beta}}_n)^\top \mathbf{K}_n^{-1}(\mathbf{y}_n - \mathbf{P}_n\hat{\boldsymbol{\beta}}_n), \tag{3.2.6}$$

then plug-in this into the acquisition function (3.2.5). This gives the following plug-in estimate of the EI acquisition function:

$$\mathrm{EI}_n(\mathbf{x}) = \underbrace{I_n(\mathbf{x})\Phi\left(\frac{I_n(\mathbf{x})}{\hat{\sigma}_n s_n(\mathbf{x})}\right)}_{\textbf{Exploitation}} + \underbrace{\hat{\sigma}_n s_n(\mathbf{x})\phi\left(\frac{I_n(\mathbf{x})}{\hat{\sigma}_n s_n(\mathbf{x})}\right)}_{\textbf{Exploration}}. \tag{3.2.7}$$

With (3.2.7) in hand, the next query point $\mathbf{x}_{n+1}$ is obtained by maximizing the EI acquisition function $\mathrm{EI}_n(\mathbf{x})$, i.e.:

$$\mathbf{x}_{n+1} \leftarrow \underset{\mathbf{x}\in\Omega}{\operatorname{argmax}}\, \mathrm{EI}_n(\mathbf{x}).$$

This maximization of (3.2.7) implicitly captures the aforementioned *exploration-exploitation trade-off*: exploration of the feasible region and exploitation near the current best solution. The maximization of the first term in (3.2.7) encourages *exploitation*, since larger values are assigned for points $\mathbf{x}$ with smaller predicted values $\hat{f}_n(\mathbf{x})$. The maximization of the second term in (3.2.7) encourages *exploration*, since larger values for points $\mathbf{x}$ indicate larger (estimated) predictive standard deviation $\hat{\sigma}_n s_n(\mathbf{x})$.

However, one drawback of EI is that it fails to capture the full uncertainty of model parameters within the acquisition function $\mathrm{EI}_n(\mathbf{x})$. This results in an *over-exploitation* of the fitted GP model for optimization, and an *under-exploration* of the black-box function over $\Omega$. This over-greediness has been noted in several works, e.g., [69] and [68]. In particular, Theorem 3 of [69] showed that, for a common class of correlation functions for $K$ (see Assumption 3.5.1 later), there always exists some smooth function $f$ within a function space $\mathcal{H}_{\boldsymbol{\theta}}(\Omega)$ (defined later in Section 3.5) such that EI fails to find a global optimum of $f$. This is stated formally below:

**Proposition 3.2.1** (Theorem 3, [69])**.** Suppose Assumption 3.5.1 holds with $\nu < \infty$. Suppose initial points are sampled according to some probability measure $F$ over $\Omega$. Let $(\mathbf{x}_i)_{i=1}^{\infty}$ be the points generated by maximizing $\text{EI}_n$ in (3.2.7). Then, for any $\epsilon > 0$, there exist some $f \in \mathcal{H}_{\boldsymbol{\theta}}(\Omega)$ and some constant $\delta > 0$ such that

$$\mathbb{P}_F\left(\lim_{n\to\infty} y_n^* - \min_{\mathbf{x}\in\Omega} f(\mathbf{x}) \geq \delta\right) > 1 - \epsilon.$$

This proposition shows that, even for relatively simple objective functions $f$, the EI may fail to converge to a global minimum due to its over-exploitation of the fitted GP model. This is concerning not only from an asymptotic perspective, but also in practical problems with limited samples. As we show later, this overexploitation can cause the EI to get stuck on suboptimal solutions, resulting in poor empirical performance compared to the HEI.

## 3.3 Hierarchical Expected Improvement

To address this, we propose a hierarchical EI framework, which aims to provide a richer quantification of parameter uncertainty within the acquisition function. The key ingredient in HEI is a *hierarchical* GP model on $f(\mathbf{x})$. Let us adopt the universal kriging model (3.2.1), but now with the following hierarchical priors assigned on parameters $(\boldsymbol{\beta}, \sigma^2)$:

$$[\boldsymbol{\beta}] \propto \mathbf{1}, \qquad [\sigma^2] \sim \text{IG}(a, b). \tag{3.3.1}$$

In words, the coefficients $\boldsymbol{\beta}$ are assigned a flat improper (*i.e.*, non-informative) prior over $\mathbb{R}^q$, and the process variance $\sigma^2$ is assigned a conjugate inverse-Gamma prior with shape and scale parameters $a$ and $b$, respectively. The idea is to leverage this hierarchical structure on model parameters to account for estimation uncertainty, while preserving a closed-form criterion for efficient sequential sampling.

The next lemma provides the posterior distribution of $f(\mathbf{x})$ under this hierarchical model:

**Lemma 3.3.1.** Assume the universal kriging model (3.2.1) with hierarchical priors (3.3.1) and $n > q$. Given data $\mathcal{D}_n$, we have

$$\left[\sigma^2 \middle| \mathcal{D}_n\right] \sim \mathrm{IG}\big(a_n, b_n\big) \quad \text{and} \quad \left[\boldsymbol{\beta} \middle| \mathcal{D}_n\right] \sim T_q\big(2a_n, \hat{\boldsymbol{\beta}}_n, \tilde{\sigma}_n^2 \mathbf{G}_n^{-1}\big), \tag{3.3.2}$$

where $a_n = a + (n-q)/2$, $b_n = b + n\hat{\sigma}_n^2/2$, $\tilde{\sigma}_n^2 = b_n/a_n$, and $T_q(\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a $q$-dimensional non-standardized t-distribution with degrees of freedom $\nu$, location vector $\boldsymbol{\mu}$ and scale matrix $\boldsymbol{\Sigma}$. Furthermore, the posterior distribution of $f(\mathbf{x})$ is

$$\left[f(\mathbf{x}) \middle| \mathcal{D}_n\right] \sim T_1\big(2a + n - q, \hat{f}_n(\mathbf{x}), \tilde{\sigma}_n s_n(\mathbf{x})\big). \tag{3.3.3}$$

The proof of this lemma follows from Chapter 4.4 of [59]. Lemma 3.3.1 shows that under the universal kriging model (3.2.1) with hierarchical priors (3.3.1), the posterior distribution of $f(\mathbf{x})$ is now a non-standarized $t$-distribution, with closed-form expressions for its location and scale parameters $\hat{f}_n(\mathbf{x})$ and $\tilde{\sigma}_n s_n(\mathbf{x})$.

Comparing the predictive distributions in (3.2.2) and (3.3.3), there are several differences which highlight the increased predictive uncertainty from the hierarchical GP model. First, the new posterior (3.3.3) is now $t$-distributed, whereas the earlier posterior (3.2.2) is normally distributed. This suggests that the hierarchical model imposes heavier tails, which increases predictive uncertainty. Second, the variance term $\tilde{\sigma}_n^2$ in (3.3.3) can be decomposed as:

$$\tilde{\sigma}_n^2 = (2b + n\hat{\sigma}_n^2)/(2a + (n-q)) > n/(2a + (n-q)) \cdot \hat{\sigma}_n^2. \tag{3.3.4}$$

When $a < q/2$ (which is satisfied via a weakly informative prior on $\sigma^2$), $\tilde{\sigma}_n^2$ is larger than the MLE $\hat{\sigma}_n^2$, which again increases predictive uncertainty.

Similar to the EI criterion (3.2.5), we now define the HEI acquisition function as:

$$\mathrm{HEI}_n(\mathbf{x}) = \mathbb{E}_{f|\mathcal{D}_n}(y_n^* - f(\mathbf{x}))_+, \tag{3.3.5}$$

where the conditional expectation over $[f(\mathbf{x})|\mathcal{D}_n]$ is under the hierarchical GP model. The proposition below gives a *closed-form* expression for $\mathrm{HEI}_n(\mathbf{x})$:

**Proposition 3.3.2.** Assume the universal kriging model (3.2.1) with hierarchical priors (3.3.1) and $n > q$. Then:

$$\mathrm{HEI}_n(\mathbf{x}) = \underbrace{I_n(\mathbf{x})\Phi_{\nu_n}\left(\frac{I_n(\mathbf{x})}{\tilde{\sigma}_n s_n(\mathbf{x})}\right)}_{\textbf{Exploitation}} + \underbrace{m_n\tilde{\sigma}_n s_n(\mathbf{x})\phi_{\nu_n-2}\left(\frac{I_n(\mathbf{x})}{m_n\tilde{\sigma}_n s_n(\mathbf{x})}\right)}_{\textbf{Exploration}}, \tag{3.3.6}$$

where $m_n = \sqrt{\nu_n/(\nu_n - 2)}$, $\nu_n = 2a_n$, and $\phi_{\nu_n}(x)$, $\Phi_{\nu_n}(x)$ denote the p.d.f. and c.d.f. of a Student's t-distribution with $\nu_n$ degrees of freedom, respectively.

Proposition 3.3.2 shows that the HEI criterion preserves the desirable properties of original EI criterion (3.2.7): it has an easily-computable, closed-form expression, which allows for efficient optimization of the next query point. This HEI criterion also has an equally interpretable exploration-exploitation trade-off. Similar to the EI criterion, the first term encourages exploitation near the current best solution $\mathbf{x}_n^*$, and the second term encourages exploration of regions with high predictive variance.

More importantly, the differences between the HEI (3.3.6) and the EI (3.2.7) acquisition functions highlight how our approach addresses the over-greediness issue. There are three notable differences. First, the HEI exploration term depends on the $t$-p.d.f. $\phi_{\nu_n-2}$, whereas the EI exploration term depends on the normal p.d.f. $\phi$. Since the former has heavier tails, the HEI exploration term is inflated, which encourages exploration. Second, the larger variance term $\tilde{\sigma}_n^2$ (see (3.3.4)) also inflates the HEI exploration term and encourages exploration. Third, the HEI contains an additional adjustment factor $\sqrt{\nu_n/(\nu_n-2)}$ in its exploration term. Since this factor is larger than 1, HEI again encourages exploration. This adjustment is most prominent for small sample sizes, since the factor $\sqrt{\nu_n/(\nu_n-2)} \to 1$ as sample size $n \to \infty$. All three differences *correct* the over-exploitation of EI via a principled hierarchical Bayesian framework. We will show later that these modifications for the HEI address the aforementioned theoretical and empirical limitations of the EI.

Finally, we note that the Student EI, proposed by [73], can be viewed as a special case of the HEI criterion, with (i) a constant mean function $\mu(\mathbf{x}) = \mu$, and (ii) "fixed" hyperparameters $a$ and $b$ (in that it does not scale with sample size $n$) for the inverse-Gamma prior in (3.3.1). We will show later that the HEI, by generalizing (i) and (ii), can yield improved theoretical and empirical performance over the SEI. For (i), instead of the *stationary* GP model used in the SEI, the HEI instead considers a broader *non-stationary* GP model with mean function $\mu(\mathbf{x}) = \mathbf{p}^\top(\mathbf{x})\boldsymbol{\beta}$, and factors in the uncertainty on coefficients $\boldsymbol{\beta}$ for optimization. This allows HEI to integrate uncertainty on GP nonstationarity to encourage more exploration in sequential sampling. For (ii), [73] recommended a "fixed" hyperparameter setting for the SEI, where the hyperparameters $a$ and $b$ do not scale with sample size $n$. However, the following proposition shows that the SEI (under such a setting) can fail to find the global optimum, under mild regularity conditions (see Assumptions 3.5.1 and 3.5.2 later).

**Proposition 3.3.3.** Suppose Assumptions 3.5.1 and 3.5.2 hold with $\nu < \infty$. Suppose initial points are sampled according to some probability measure $F$ over $\Omega$. Given fixed hyperparameters $a$ and $b$, let $(\mathbf{x}_i)_{i=1}^\infty$ be the points returned by the SEI procedure. Then, for any $\epsilon > 0$, there exist some $f \in \mathcal{H}_{\boldsymbol{\theta}}(\Omega)$ and some constant $\delta > 0$ such that

$$\mathbb{P}_F\left(\lim_{n\to\infty} y_n^* - \min_{\mathbf{x}\in\Omega} f(\mathbf{x}) \geq \delta\right) > 1-\epsilon.$$

The proof is provided in Appendix C.1.2. This proposition shows that the SEI with "fixed" hyperparameters has the same limitation as the EI: it can fail to converge to a global minimum for relatively smooth objective functions $f$. We will show later in Section 3.5 that, under a more general prior specification which allows the hyperparameter $b$ to scale with sample size $n$ (more specifically, $b = \Theta(n)$), the HEI not only has the desired global convergence property for optimization, but also a near-minimax convergence rate.

## 3.4 Methodology and Algorithm

Using the HEI acquisition function (3.3.6), we now present a methodology for integrating this for effective black-box optimization. We first introduce hyperparameter estimation techniques which allow the HEI to mimic a fully Bayesian optimization procedure, then present an algorithmic framework for implementing the HEI.

### 3.4.1 Hyperparameter Specification

We present below several plausible specifications for the hyperparameters $(a, b)$ in the hierarchical prior $[\sigma^2] \sim \mathrm{IG}(a, b)$ in (3.3.1), and discuss when certain specifications may yield better optimization performance.

**(i) Weakly Informative.** Consider first a *weakly informative* specification of the hyperparameters $(a, b)$, with $a = b = \epsilon$ for a small choice of $\epsilon$, *e.g.*, $\epsilon = 0.1$. This reflects weak information on the variance parameter $\sigma^2$, and provides regularization for parameter inference. The limiting case of $\epsilon \to 0$ yields the non-informative Jeffreys' prior for variance parameters.

While weakly informative (and non-informative) priors are widely used in Bayesian analysis [76], we have found that such priors can result in poor optimization performance for HEI (Section 3.6 provides further details). One reason is that, for many black-box problems, only a small sample size can be afforded on the objective function $f$, since each evaluation is expensive. One can perhaps address this with a carefully elicited subjective prior, but such informative priors are typically not available when the objective $f$ is black-box. We present next two specifications which may offer improved optimization performance, both in theory (Section 3.5) and in practice (Sections 3.6 and 3.7).

**(ii) Empirical Bayes.** Consider next an empirical Bayes (EB, [77]) approach, which uses the observed data on $f$ to estimate the hyperparameters $(a, b)$. This is achieved by maximizing the following marginal likelihood for $(a, b)$:

$$p(\mathbf{y}_n; a, b) = \int \mathcal{L}(\boldsymbol{\beta}, \sigma^2; \mathbf{y}_n)\pi(\boldsymbol{\beta})\pi(\sigma^2; a, b)\ d\boldsymbol{\beta} d\sigma^2. \tag{3.4.1}$$

Here, $\mathcal{L}(\boldsymbol{\beta}, \sigma^2; \mathbf{y}_n)$ is the likelihood function of the universal kriging model (3.2.1) (see [59] for the full expression), and $\pi(\boldsymbol{\beta})$ and $\pi(\sigma^2; a, b)$ are the prior densities of $\boldsymbol{\beta}$ and $\sigma^2$ given hyperparameters $a$ and $b$. The model with estimated hyperparameters via EB provides a close approximation to a fully hierarchical Bayesian model [77], where additional hyperpriors are assigned on $a$ and $b$. The latter can be viewed as a "gold standard" quantification of model uncertainty, but typically requires MCMC sampling, which can be more expensive than optimization. Here, an EB estimate of hyperparameters $(a, b)$ would allow the HEI to closely *mimic* a fully Bayesian optimization procedure (the "gold standard"), while avoiding expensive MCMC sampling via a *closed-form* acquisition function.

Unfortunately, the proposition below shows that a direct application of EB for the HEI yields unbounded hyperparameter estimates:

**Proposition 3.4.1.** The marginal likelihood for the universal kriging model (3.2.1) with priors (3.3.1) is given by:

$$p(\mathbf{y}_n; a, b) = \det(\mathbf{G}_n \mathbf{K}_n)^{-\frac{1}{2}} \frac{b^a}{\Gamma(a)} \frac{\Gamma(a + (n-q)/2)}{(b + w_n)^{a + \frac{n-q}{2}}}, \tag{3.4.2}$$

where $w_n = (\mathbf{y}_n^\top \mathbf{K}_n^{-1} \mathbf{y}_n - \hat{\boldsymbol{\beta}}_n^\top \mathbf{G}_n \hat{\boldsymbol{\beta}}_n)/2$. Furthermore, the maximization problem:

$$\underset{a>0,b>0}{\text{argmax}} \; p(\mathbf{y}_n; a, b) \tag{3.4.3}$$

is unbounded for all values of $\mathbf{y}_n$.

The proof of Proposition 3.4.1 is provided in Appendix C.2.1.

To address this issue of unboundedness, one can instead use a modified EB approach, called the marginal maximum a posteriori estimator (MMAP, [78]). The MMAP adds an additional level of hyperpriors $\pi(a, b)$ to the marginal likelihood maximization problem, yielding the modified formulation:

$$\underset{a>0,b>0}{\text{argmax}} \; \tilde{p}(\mathbf{y}_n; a, b) := p(\mathbf{y}_n; a, b)\pi(a, b). \tag{3.4.4}$$

The MMAP approach for hyperparameter specification has been used in a variety of problems, *e.g.*, scalable training of large-scale Bayesian networks [79]. The next proposition shows that the MMAP indeed yields finite solutions for a general class of hyperpriors on $(a, b)$:

**Proposition 3.4.2.** Assume the following independent hyperpriors on $(a, b)$:

$$[a] \sim \text{Gamma}(\zeta, \iota), \qquad [b] \propto \mathbf{1}, \tag{3.4.5}$$

where $\zeta$ and $\iota$ are the shape and scale parameters, respectively. Then the maximization of $\tilde{p}(\mathbf{y}_n; a, b)$ is always finite for $(a, b) \in \mathbb{R}_+^2$.

The proof of Proposition 3.4.2 is provided in Appendix C.2.2. By mimicking a fully Bayesian optimization procedure, this MMAP approach can consistently outperform the weakly informative specification for HEI – we will show this later in numerical studies.

**(iii) Data-Size-Dependent (DSD).** Finally, we consider the so-called "data-size-dependent" (DSD) hyperparameter specification. This is motivated from the prior specification needed for global optimization convergence of the HEI, which we present and justify in the following section. The DSD specification requires the shape parameter $a$ to be constant, and the scale parameter $b$ to grow at the same order as the sample size $n$, *i.e.*, $b = \kappa n$ for some constant $\kappa > 0$. One appealing property of this specification is that it ensures the HEI converges to a global optimum $\mathbf{x}^*$ (see Theorem 3.5.3 later).

For the DSD specification, we can similarly use the MMAP to estimate hyperparameters $(a, \kappa)$, to mimic a fully Bayesian EI procedure. Suppose data $\mathcal{D}_{n_{\text{ini}}}$ are collected from $n_{\text{ini}}$ initial design points (more on this in Section 3.4.3). Then the hyperparameters $a$ and $\kappa$ can be estimated via the MMAP optimization:

$$(a^*, \kappa^*) = \underset{a>0, \kappa>0}{\text{argmax}} \left\{ p(\mathbf{y}_{n_{\text{ini}}}; a, \kappa n_{\text{ini}}) \pi(a, \kappa) \right\}, \tag{3.4.6}$$

where $\pi(a, \kappa)$ is the hyperprior density on $a$ and $\kappa$. One possible setting for $\pi(a, \kappa)$ is a Gamma hyperprior on $a$ and a non-informative hyperprior $[\kappa] \propto 1$ (independent of $a$). By Proposition 6, this specification again yields a finite optimization problem for MMAP. Using these estimated hyperparameters, subsequent points are then queried using HEI with $a = a^*$ and $b = \kappa^* n$, where $n$ is the current sample size.

### 3.4.2 Order Selection for Basis Functions

In addition to hyperparameter estimation, the choice of basis functions in $\mathbf{p}(\mathbf{x})$ and the order selection of such bases are also important for an effective implementation of the HEI. In our experiments, we take these bases to be complete polynomials up to a certain order $l$. Letting $\mathcal{M}^{(l)}$ denote the polynomial model with maximum order $l$, we have $\mathbf{p}(\mathbf{x}) = 1$ for model $\mathcal{M}^{(0)}$ (a constant model), $\mathbf{p}(\mathbf{x}) = [1, x_1, \cdots, x_d]^\top$ for model $\mathcal{M}^{(1)}$ (a linear model), $\mathbf{p}(\mathbf{x}) = [1, x_1, \cdots, x_d, x_1^2, \cdots, x_d^2, x_1x_2, \cdots, x_1x_d, x_2x_3, \cdots, x_{d-1}x_d]^\top$ for model $\mathcal{M}^{(2)}$ (a second-order interaction model), *etc*. One can also make use of other basis functions (*e.g.*, orthogonal polynomials; [80]) depending on the problem at hand.

A careful selection of order $l$ is also important: an overly small estimate of $l$ results in over-exploitation of a poorly-fit model, whereas an overly large estimate results in variance inflation and over-exploration of the domain. We found that the standard Bayesian Information Criterion (BIC) [81] provides good order selection performance for the HEI. Given initial data $\mathcal{D}_{n_{\text{ini}}}$, the BIC selects the model $\mathcal{M}^{(l^*)}$ with order:

$$l^* = \underset{l \in \mathbb{N}}{\operatorname{argmin}} \big\{ -2 \log \mathcal{L}(\mathcal{M}^{(l)}) + q_l \log(n_{\text{ini}}) \big\}. \tag{3.4.7}$$

Here, $\mathcal{L}(\mathcal{M}^{(l)})$ denotes the likelihood of model $\mathcal{M}^{(l)}$ (this likelihood expression can be found in [59]), and $q_l$ denotes the number of basis functions in model $\mathcal{M}^{(l)}$. With this optimal order selected, subsequent samples are then obtained using HEI with mean function following this polynomial order.

### 3.4.3 Algorithm Statement

**Algorithm 1** Hierarchical Expected Improvement for Bayesian Optimization

**Initialization**

- Generate $n_{\text{ini}}$ space-filling design points $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n_{\text{ini}}}\}$ on $\Omega$.
- Evaluate function points $y_i = f(\mathbf{x}_i)$, yielding initial points $\mathcal{D}_{n_{\text{ini}}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_{\text{ini}}}$.

**Model selection**

- Select model order via BIC using (3.4.7).
- Estimate hyperparameters $(a, b)$ via MMAP using (3.4.4).

**Optimization**

**for** $n \leftarrow n_{\text{ini}}$ *to* $n_{\text{tot}} - 1$ **do**

Given $\mathcal{D}_n$, estimate length-scale parameters $\boldsymbol{\theta}$ via MAP and get $\text{HEI}_n(\mathbf{x})$.

Obtain the next evaluation point $\mathbf{x}_{n+1}$ by maximizing $\text{HEI}_n(\mathbf{x})$:

$$\mathbf{x}_{n+1} \leftarrow \underset{\mathbf{x} \in \Omega}{\operatorname{argmax}} \, \text{HEI}_n(\mathbf{x}).$$

Evaluate $y_{n+1} = f(\mathbf{x}_{n+1})$, and update data $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \big\{(\mathbf{x}_{n+1}, y_{n+1})\big\}$.

**Return:** The best observed solution $\mathbf{x}_{i^*}$, where $i^* = \operatorname{argmin}_{i=1}^{n_{\text{tot}}} f(\mathbf{x}_i)$.

Algorithm 1 summarizes the above steps for HEI. First, initial data on the black-box function $f$ are collected on a "space-filling" design, which provides good coverage of the feasible space $\Omega$. For the unit hypercube $\Omega = [0,1]^d$, we have found that the maximin Latin hypercube design (MmLHD, [82]) works quite well in practice. For non-hypercube domains, more elaborate design methods on non-hypercube regions (*e.g.*, [83, 84, 85]) can be used. The number of initial points is set as $n_{\text{ini}} = 10d$, as recommended in [86]. Using this initial data, the model order for the hierarchical GP is selected using (3.4.7). The hyperparameters $a$ and $b$ are also estimated from data (if necessary) using the methods described in Section 3.4.1. Next, the following two steps are repeated until the sample size budget $n_{\text{tot}}$ is exhausted: (i) the GP length-scale parameters $\boldsymbol{\theta}$ are fitted via maximum a posteriori (MAP) estimation[1] using the observed data points, (ii) a new sample $f(\mathbf{x})$ is collected at the point $\mathbf{x}$ which maximizes the HEI criterion (3.3.6).

### 3.5 Convergence Analysis

We present next the global optimization convergence result for the HEI, then provide a near-minimax optimal convergence rate for the proposed method. In what follows, we will assume that the domain $\Omega$ is convex and compact.

Let us first adopt the following shift-invariant form for the kernel $K$:

$$K_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) := C\left(\frac{x_1 - z_1}{\theta_1}, \ldots, \frac{x_d - z_d}{\theta_d}\right), \tag{3.5.1}$$

where $C$ is a stationary correlation function with $C(\mathbf{0}) = 1$ and length-scale parameters $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_d)$. From this, we can then define a function space – the reproducing kernel Hilbert space (RKHS, [87]) – for the objective function $f$. Given kernel $K_{\boldsymbol{\theta}}$ (which is

[1] In numerical experiments, we use an independent uniform prior $\theta_l \overset{\text{i.i.d.}}{\sim} U[0, 100]$ for this MAP estimate.

symmetric and positive definite), define the linear space

$$\mathcal{F}_{\boldsymbol{\theta}}(\Omega) = \left\{ \sum_{i=1}^{N} \alpha_i K_{\boldsymbol{\theta}}(\cdot, \mathbf{x}_i) : N \in \mathbb{N}_+, \mathbf{x}_i \in \Omega, \alpha_i \in \mathbb{R} \right\}, \tag{3.5.2}$$

and equip this space with the bilinear form

$$\left\langle \sum_{i=1}^{N} \alpha_i K_{\boldsymbol{\theta}}(\cdot, \mathbf{x}_i), \sum_{j=1}^{M} \gamma_j K_{\boldsymbol{\theta}}(\cdot, \mathbf{y}_j) \right\rangle_{K_{\boldsymbol{\theta}}} := \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_i \gamma_j K_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{y}_j). \tag{3.5.3}$$

The RKHS $\mathcal{H}_{\boldsymbol{\theta}}(\Omega)$ of kernel $K_{\boldsymbol{\theta}}$ is defined as the closure of $\mathcal{F}_{\boldsymbol{\theta}}(\Omega)$ under $\langle \cdot, \cdot \rangle_{K_{\boldsymbol{\theta}}}$, with its inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_{\boldsymbol{\theta}}}$ induced by $\langle \cdot, \cdot \rangle_{K_{\boldsymbol{\theta}}}$.

Next, we make the following two regularity assumptions. The first is a smoothness assumption on the correlation function $C$:

**Assumption 3.5.1.** $C$ is continuous, integrable, and satisfies:

$$|C(\mathbf{x}) - Q_r(\mathbf{x})| = \mathcal{O}\left( \|\mathbf{x}\|_2^{2\nu} (-\log \|\mathbf{x}\|_2)^{2\alpha} \right) \quad \text{as} \quad \|\mathbf{x}\|_2 \to 0,$$

for some constants $\nu > 0$ and $\alpha \geq 0$. Here, $r = \lfloor 2\nu \rfloor$ and $Q_r(\mathbf{x})$ is the $r$-th order Taylor approximation of $C(\mathbf{x})$. Furthermore, its Fourier transform

$$\widehat{C}(\boldsymbol{\xi}) := \int_{\mathbf{x}} e^{-2\pi i \langle \boldsymbol{\xi}, \mathbf{x} \rangle} C(\mathbf{x}) d\mathbf{x}$$

is isotropic, radially non-increasing and satisfies either: as $\|\mathbf{x}\|_2 \to \infty$

$$\widehat{C}(\mathbf{x}) = \Theta\left( \|\mathbf{x}\|_2^{-2\nu - d} \right) \quad \text{or} \quad \widehat{C}(\mathbf{x}) = \mathcal{O}\left( \|\mathbf{x}\|_2^{-2\lambda - d} \right) \quad \text{for any } \lambda > 0.$$

Note that in Assumption 3.5.1, since we assume $C$ is continuous and integrable, its Fourier transform $\hat{C}$ must exist. A widely-used correlation function which satisfies this assumption is the Matérn correlation function (see [88] and [69]).

The second assumption is a regularity condition on the MAP estimator for the length-scale parameters $\boldsymbol{\theta}$.

**Assumption 3.5.2.** Given data $\mathcal{D}_n$, let $\tilde{\boldsymbol{\theta}}_n$ be the MAP of $\boldsymbol{\theta}$ under prior $\pi(\boldsymbol{\theta})$. For any $n > q$, we assume that

$$\boldsymbol{\theta}^L \leq \tilde{\boldsymbol{\theta}}_n \leq \boldsymbol{\theta}^U \quad \text{for some constants } \boldsymbol{\theta}^L, \boldsymbol{\theta}^U \in \mathbb{R}_+^d. \tag{3.5.4}$$

Under these two regularity assumptions, we can then prove the global optimization convergence of the HEI method (more specifically, for HEI-DSD).

**Theorem 3.5.3.** Suppose Assumptions 3.5.1 and 3.5.2 hold. Further suppose the hyperparameter $a$ is a constant (in $n$) and $b = \Theta(n)$, with basis functions $p_i(x) \in \mathcal{H}_{\theta^U}(\Omega)$. Let $(\mathbf{x}_i)_{i=1}^{\infty}$ be the points generated by maximizing HEI$_n$ in (3.3.6), with iterative plug-in MAP estimates $\tilde{\boldsymbol{\theta}}_n$. Then, for any $f \in \mathcal{H}_{\boldsymbol{\theta}^U}(\Omega)$ and any choice of initial points $\{\mathbf{x}_i\}_{i=1}^{n_{\rm ini}}$, we have:

$$y_n^* - \min_{\mathbf{x}\in\Omega} f(\mathbf{x}) = \begin{cases} \mathcal{O}(n^{-\nu/d}(\log n)^{\alpha}), & \nu \leq 1, \\ \mathcal{O}(n^{-1/d}), & \nu > 1. \end{cases} \tag{3.5.5}$$

The proof of this theorem is given in Appendix C.3.1. The key idea is to upper bound the prediction gap $f(\mathbf{x}) - \hat{f}_n(\mathbf{x})$ by the posterior variance term $s_n^2(\mathbf{x})$ in (3.2.4), which is a generalization of the power function used in the function approximation literature (see, *e.g.*, Theorem 11.4 of [87]). We then show that the hyperparameter assumption $b = \Theta(n)$ prevents the variance estimate $\tilde{\sigma}_n^2$ from collapsing to 0, and allows us to apply approximation bounds on $s_n^2(\mathbf{x})$ to obtain the desired global convergence result. This proof is inspired by Theorem 4 of [69].

Theorem 3.5.3 shows that, for all objective functions $f$ in the RKHS $\mathcal{H}_{\boldsymbol{\theta}^U}(\Omega)$, the HEI indeed has the desired convergence property for global optimization. This addresses the lack of convergence for the EI from Proposition 3.2.1 and the SEI from Proposition 3.3.3. It is worth nothing that, when $C$ is the Matérn correlation with smoothness parameter $\nu$ [88], the RKHS $\mathcal{H}_{\boldsymbol{\theta}}(\Omega)$ consists of functions $f$ with continuous derivatives of order $\nu' < \nu$ [59]. Hence, for the Matérn correlation, the HEI achieves a global convergence rate of $\mathcal{O}(n^{-1/d})$ for objective functions $f \in \mathcal{H}_{\boldsymbol{\theta}}(\Omega)$ with $\nu > 1$, and $\mathcal{O}(n^{-\nu/d}(\log n)^{\alpha})$ for objective functions $f \in \mathcal{H}_{\boldsymbol{\theta}}(\Omega)$ with $\nu \leq 1$.

At first glance, the prior specification in Theorem 3.5.3 may appear slightly peculiar, since the hyperparameter $b = \Theta(n)$ depends on the sample size $n$. However, such *data-size-dependent* priors have been studied extensively in the context of high-dimensional Bayesian linear regression, particularly in its connection to optimal minimax estimation (see, *e.g.*, [89]). The data-size-dependent prior in Theorem 3.5.3 can be interpreted in a similar way: the hyperparameter condition $b = \Theta(n)$ is sufficient in encouraging exploration in the sequential sampling points, so that HEI converges to a global optimum for all $f$ in the RKHS $\mathcal{H}_{\boldsymbol{\theta}^U}(\Omega)$.

Under a further $\gamma$-stability assumption [90], we can further show that the HEI achieves a near-minimax convergence rate for Bayesian optimization (we provide discussions on this minimax rate at the end of the section). This assumption is stated below:

**Assumption 3.5.4.** Let $(\mathbf{x}_i)_{i=1}^{\infty}$ be the sequence of points generated by the HEI. We assume that

$$s_n(\mathbf{x}_{n+1}) \geq \gamma \|s_n(\mathbf{x})\|_{\infty} \quad \text{for all } n = 1, 2, \cdots, \tag{3.5.6}$$

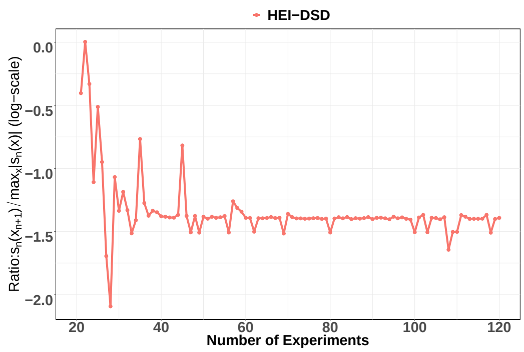
for some constant $\gamma \in (0, 1]$.

In words, this assumes that every sequential point $\mathbf{x}_{n+1}$ has a posterior standard deviation term $s_n(\mathbf{x}_{n+1})$ (from (3.2.4)) at least as large as $\gamma \|s_n(\mathbf{x})\|_{\infty}$, where $\|s_n(\mathbf{x})\|_{\infty}$ is the maximum posterior standard deviation term over domain $\Omega$. Under this assumption, we can then prove the desired convergence rate:

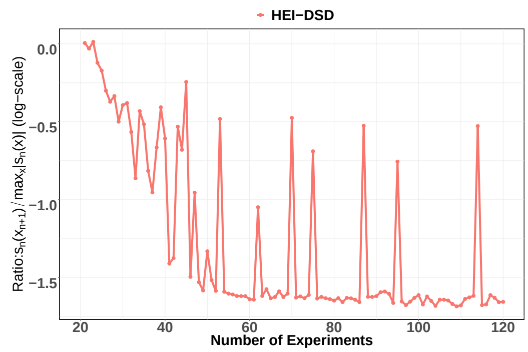
**Theorem 3.5.5.** Suppose Assumptions 3.5.1, 3.5.2 and 3.5.4 hold. Further suppose the hyperparameter $a$ is a constant (in $n$) and $b = \Theta(n)$, with basis functions $p_i(x) \in \mathcal{H}_{\theta^U}(\Omega)$. Let $(\mathbf{x}_i)_{i=1}^{\infty}$ be the points generated by maximizing $\text{HEI}_n$ in (3.3.6), with iterative plug-in MAP estimates $\tilde{\boldsymbol{\theta}}_n$ and constraint (3.5.6). Then, for any $f \in \mathcal{H}_{\boldsymbol{\theta}^U}(\Omega)$ and any initial points, we have:

$$y_n^* - \min_{\mathbf{x}\in\Omega} f(\mathbf{x}) = O(n^{-\nu/d+1/2}). \tag{3.5.7}$$

The proof of this theorem is provided in Appendix C.3.2.




(a) Branin Function.




(b) Three-Hump Camel Function.

Figure 3.1: The log-ratio $\log\{s_n(\mathbf{x}_{n+1})/\|s_n(\mathbf{x})\|_\infty\}$ for HEI-DSD using the Branin and Three-Hump Camel test functions, as a function of sample size $n$.

One way to ensure Assumption 3.5.4 holds is to explicitly impose the constraint in (3.5.6) when optimizing the HEI acquisition function. This requires a small modification of Algorithm 1 and is relatively easy to implement. However, in the numerical experiments later, this assumption appears to be satisfied empirically without any modifications needed on Algorithm 1. Figure 3.1 shows the log-ratio $\log\{s_n(\mathbf{x}_{n+1})/\|s_n(\mathbf{x})\|_\infty\}$ as a function of sample size $n$ for HEI-DSD, using two test functions in the later simulation study. For both functions, this ratio appears to reach a lower bound as $n$ increases, which suggests that the HEI with data-size-dependent hyperparameter specification indeed satisfies Assumption 3.5.4 for a sufficiently small $\gamma$. In such cases, the HEI-DSD would achieve the optimization rate of $O(n^{-\nu/d+1/2})$ without any need for modifying Algorithm 1.

The convergence rate in Theorem 3.5.5 for the HEI is nearly minimax for the class of objective functions considered. [69] proved that, of all optimization strategies for minimizing $f \in \mathcal{H}_\theta(\Omega)$ under Assumption 3.5.1, the minimax rate for the optimization gap $y_n^* - \min_{\mathbf{x}\in\Omega} f(\mathbf{x})$ is $\mathcal{O}(n^{-\nu/d})$, *i.e.*, there does not exist an optimization strategy with a faster asymptotic rate. The HEI rate in Theorem 3.5.5, in this sense, is nearly minimax, with an additional factor of $O(n^{1/2})$. It should be mentioned that, while these rates provides a reassuring check for HEI convergence, such asymptotic analysis does not tell the full story on the effectiveness of a Bayesian optimization method. Indeed, [69] showed that the simple (non-adaptive) strategy of optimization via a quasi-uniform sequence (see, *e.g.*, [91]) can achieve the minimax optimization rate of $\mathcal{O}(n^{-\nu/d})$. Such a strategy, however, typically performs terribly in practice and is not competitive with existing BO methods, since it is non-adaptive to previous function evaluations. The proposed HEI (as we show next) provides excellent empirical performance, while also enjoying this near-minimax convergence rate.

### 3.6 Numerical Experiments

We now investigate the numerical performance of HEI in comparison to existing BO methods, for a suite of test optimization functions. We consider the following five test functions, taken from [92]:

- **Branin** (2-dimensional function on domain $\Omega = [0,1]^2$):

$$f(\mathbf{x}) = (x_2 - 5.1/(4\pi^2) \cdot x_1^2 + 5/\pi \cdot x_1 - 6)^2 + 10(1 - 1/(8\pi))\cos(x_1) + 10,$$

- **Three-Hump Camel** (2-dimensional function on domain $\Omega = [-2,2]^2$):

$$f(\mathbf{x}) = 2x_1^2 - 1.05x_1^4 + x_1^6/6 + x_1x_2 + x_2^2,$$

- **Six-Hump Camel** (2-dimensional function on domain $\Omega = [-2,2]^2$):

$$f(\mathbf{x}) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2,$$

- **Levy Function** (6-dimensional function on domain $\Omega = [-10,10]^6$):

$$f(\mathbf{x}) = \sin^2(\pi\omega_1) + \sum_{i=1}^{5}(\omega_i - 1)^2[1 + 10\sin^2(\pi\omega_i + 1)] + (\omega_6 - 1)^2[1 + \sin^2(2\pi\omega_6)],$$

where $\omega_i = 1 + (x_i - 1)/4$ for $i = 1, \cdots, 6$,

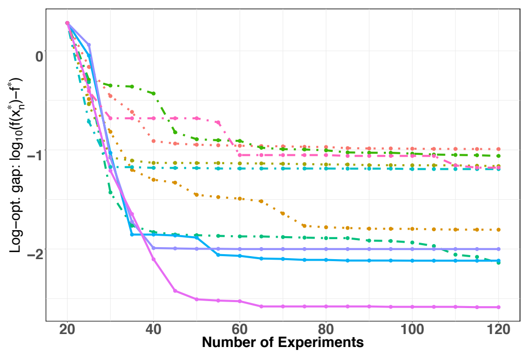
- **Ackley Function** (10-dimensional function on domain $\Omega = [-5, 5]^{10}$):

$$f(\mathbf{x}) = -20\exp\left(-\frac{0.2}{\sqrt{10}}\|\mathbf{x}\|_2\right) - \exp\left(\frac{1}{10}\sum_{i=1}^{10}\cos(2\pi x_i)\right) + 20 + \exp(1).$$

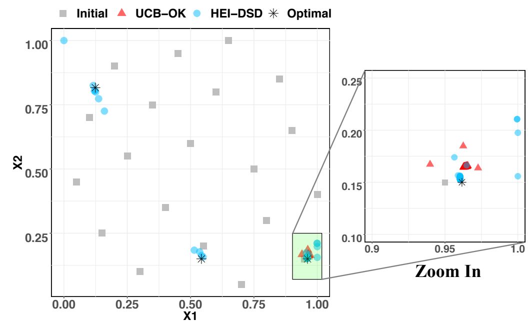
The simulation set-up is as follows. We compare the proposed HEI method under different hyperparameter specifications (HEI-Weak, HEI-MMAP, HEI-DSD), with the EI method under ordinary kriging (EI-OK) and universal kriging (EI-UK), the Student EI (SEI) method with fixed hyperparameters $(0.2, 12)$ as recommended in [73], the UCB approach under ordinary kriging (UCB-OK, [62]) with default exploration parameter $2.96$, the $\epsilon$-greedy EI approach [69] under ordinary kriging ($\epsilon$-EI-OK) and universal kriging ($\epsilon$-EI-UK) with $\epsilon = 0.1$ as suggested in [93], and the $\gamma$-stabilized EI method [90] under universal kriging (Stab-EI-UK) with $\gamma = \min(0.1d, 0.8)$[2]. For HEI-Weak, the hyperparameters $(a, b)$ are set as $a = b = 0.1$; for HEI-MMAP and HEI-DSD, the hyperparameters $(\zeta, \iota)$ are set as $\zeta = \iota = 2$. All methods use the Matérn correlation with smoothness parameter $2.5$, and are run for a total of $T = 120$ function evaluations. Here, the kriging model is fitted using the R package `kergp` [94]. All results are averaged over 20 replications.

Figures 3.2a-3.2d and 3.2f show the log-optimality gap $\log_{10}(f(\mathbf{x}_n^*) - f(\mathbf{x}^*))$ against the number of samples $n$ for the first three functions, and Figure 3.2e shows the optimality gap $f(\mathbf{x}_n^*) - f(\mathbf{x}^*)$ for the Levy function. We see that the three HEI methods outperform the existing Bayesian optimization methods: the optimality gap for the latter methods stagnates for larger sample sizes, whereas the former enjoys steady improvements as $n$ increases. This shows that the proposed method indeed corrects the over-greediness of EI,

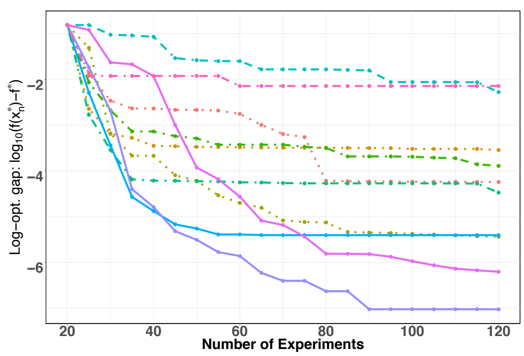
[2] Stab-EI-UK requires the next query point $\mathbf{x}_{n+1}$ to satisfy $s_n(\mathbf{x}_{n+1}) \geq \gamma\|s_n(\mathbf{x})\|_\infty$. In our implementation, we randomly sample $10^{d+2}$ points to find $\|s_n(\mathbf{x})\|_\infty$ and set $\gamma = \min(0.1d, 0.8)$ as distance increases with dimension, which achieves the near optimal convergence rate [90].



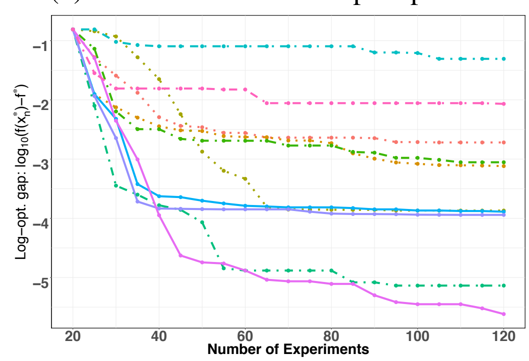

(a) Branin Function.



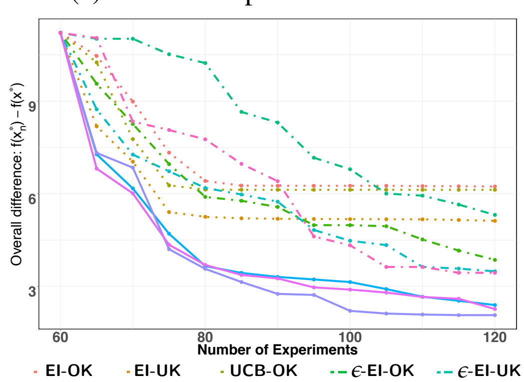

(b) A visualization of sampled points.



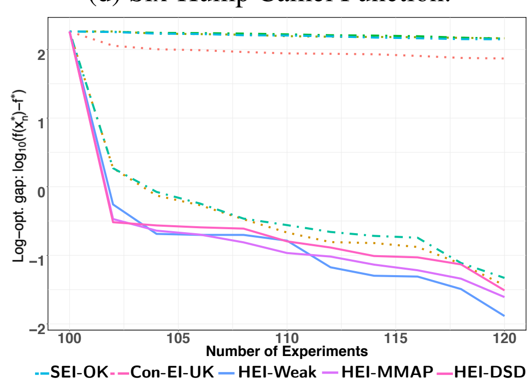

(c) Three-Hump Camel Function.




(d) Six-Hump Camel Function.




(e) Levy Function.




(f) Ackley Function.

Figure 3.2: Numerical results for synthetic functions. (a) and (c)-(f) show the average optimality gap over 10 replications (dotted lines: EI-OK, EI-UK, and UCB-OK; dashed lines: $\epsilon$-EI-OK, $\epsilon$-EI-UK, SEI-OK, Stab-EI-UK; solid lines: HEI-Weak, HEI-MMAP, HEI-DSD). (b) presents a visualization of sampled points for the Branin function (grey squares: initial points, black stars: global optima, red triangles: UCB-OK points, blue circles: HEI-DSD points).

and provides a more effective correction of this via hierarchical modeling, compared to the $\epsilon$-greedy and Stab-EI-UK methods. Furthermore, of the HEI methods, HEI-MMAP and HEI-DSD appear to greatly outperform HEI-Weak. This is in line with the earlier observation that weakly informative priors may yield poor optimization for HEI; the MMAP and DSD specifications give better performance by mimicking a fully Bayesian optimization procedure. The steady improvement of HEI-DSD also supports the data-size-dependent prior condition needed for global convergence in Theorems 3.5.3 and 3.5.5.

Figure 3.2b shows the sampled points from HEI-DSD and UCB-OK for one run of the Branin function. The points for HEI-Weak and HEI-MMAP are quite similar to HEI-DSD, and the points for EI-OK, EI-UK and SEI are quite similar to UCB-OK, so we only plot one of each for easy visualization. We see that HEI indeed encourages more exploration in optimization: it successfully finds *all* three global optima for $f$, whereas existing methods cluster points near only one optimum. The need to identify multiple global optima often arises in multiobjective optimization. For example, a company may wish to offer multiple product lines to suit different customer preferences [95]. For such problems, HEI can provide more practical solutions over existing methods.

Lastly, we compare the performance of HEI with the SEI method [73]. From Figure 3.2e, we see that the SEI performs quite well for the Levy function: it is slightly worse than HEI methods, but better than the other methods. However, from Figure 3.2, the SEI achieves only comparable performance with EI-OK for the Branin function (which is in line with the results reported in [73]), and is one of the worst-performing methods. This shows that the performance of SEI can vary greatly for different problems.

## 3.7 Semiconductor Manufacturing Optimization

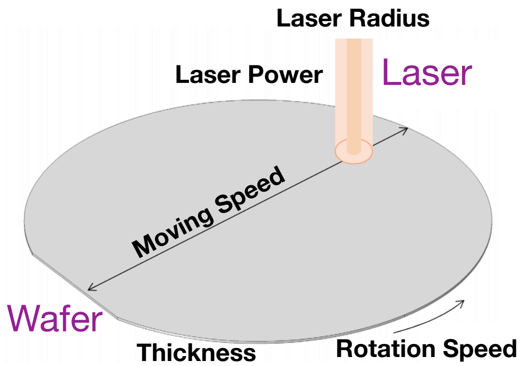
We now investigate the performance of HEI in a process optimization problem in semiconductor wafer manufacturing. In semiconductor manufacturing [96], thin silicon wafers undergo a series of refinement stages. Of these, thermal processing is one of the most important stage, since it facilitates necessary chemical reactions and allows for surface oxidation [97]. Figure 3.3a visualizes a typical thermal processing procedure: a laser beam is moved radially in and out across the wafer, while the wafer itself is rotated at a constant speed. There are two objectives here. First, the wafer should be heated to a target temperature to facilitate the desired chemical reactions. Second, temperature fluctuations over the wafer surface should be made as small as possible, to reduce unwanted strains and improve wafer fabrication [98]. The goal is to find an "optimal" setting of the manufacturing process which achieves these two objectives.

We consider five control parameters: wafer thickness, rotation speed, laser period, laser radius, and laser power (a full specification is given in Table 3.1). The heating is performed over 60 seconds, and a target temperature of $\mathcal{T}^* = 600$ F is desired over this timeframe. We use the following objective function:

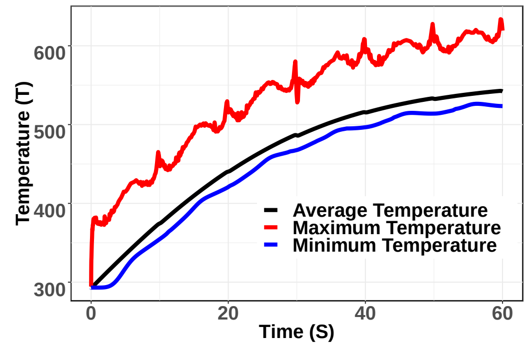
$$f(\mathbf{x}) := \sum_{t=1}^{60} \max_{\mathbf{s} \in \mathcal{S}} |\mathcal{T}_t(\mathbf{s}; \mathbf{x}) - \mathcal{T}^*|. \tag{3.7.1}$$

Here, $\mathbf{s}$ denotes a spatial location on the wafer domain $\mathcal{S}$, $t = 1, \cdots, 60$ denotes the heating time (in seconds), and $\mathcal{T}_t(\mathbf{s}; \mathbf{x})$ denotes the wafer temperature at location $\mathbf{s}$ and time $t$, using control setting $\mathbf{x} \in \mathbb{R}^5$. Note that $f(\mathbf{x})$ captures both objectives of the study: wafer temperatures $\mathcal{T}_t$ close to $\mathcal{T}^*$ results in smaller values of $f(\mathbf{x})$, and the same is true when $\mathcal{T}_t(\mathbf{s}; \mathbf{x})$ is stable over $\mathbf{s} \in \mathcal{S}$.

Clearly, each evaluation of $f(\mathbf{x})$ is expensive, since it requires a full run of wafer heating process. We will simulate each run using COMSOL Multiphysics [99], a reliable finite-element analysis software for solving complex systems of partial differential equations (PDEs). COMSOL models the incident heat flux from the moving laser as a spatially distributed heat source on the surface, then computes the transient thermal response by solving the coupled heat transfer and surface-to-ambient radiation PDEs. Figure 3.3b visualizes the simulation output from COMSOL: the average, maximum, and minimum temperature over




(a) Laser heating process of a silicon wafer.




(b) Simulated temperature profile from COMSOL Multiphysics.

Figure 3.3: A visualization of the laser heating a silicon wafer application.

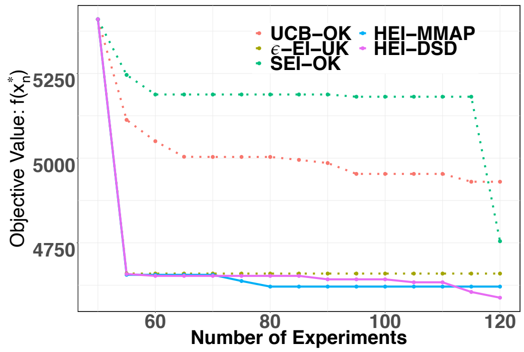
Table 3.1: Design ranges of the five control parameters, where rpm (revolutions per minute) measures the rotation speed of the wafer.

| Thickness | Rotation Speed | Laser Period | Laser Radius | Power |
|---|---|---|---|---|
| $[160, 300]\mu$s | $[2, 50]$rpm | $[5, 15]$s | $[2, 10]$mm | $[10, 20]$W |

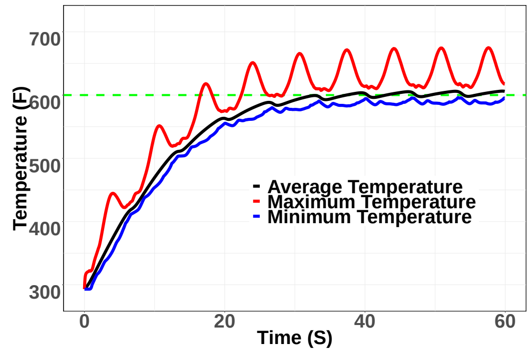
the wafer domain at every time step. Experiments are performed on a desktop computer with quad-core Intel I7-8700K processors, and take around $5$ minutes per run.

Figure 3.4a shows the best objective values $f(\mathbf{x}_n^*)$ for HEI-MMAP and HEI-DSD (the best performing HEI methods from simulations), and for the UCB-OK, SEI, and $\epsilon$-greedy EI methods. We see that UCB-OK and SEI perform noticeably poorly, whereas the proposed HEI-MMAP and HEI-DSD methods provide the best optimization performance, with the $\epsilon$-greedy-EI method slightly worse. This again shows that the proposed HEI can provide a principled correction to the over-greediness of EI via hierarchical modeling, which translates to more effective optimization performance over the compared existing methods.

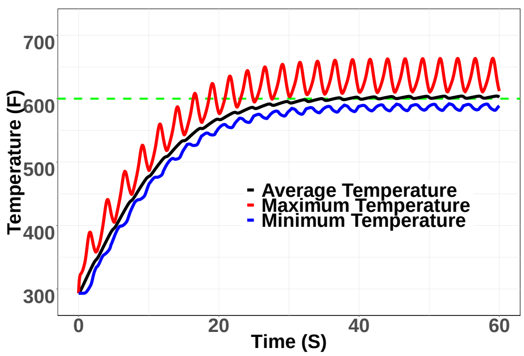
The remaining plots in Figure 3.4 show the average, maximum, and minimum temperature over the wafer surface, as a function of time. For HEI-DSD and HEI-MMAP, the average temperature quickly hits 600 F, with a slight temperature oscillation over the wafer. For SEI, the average temperature reaches the target temperature slowly, but the temperature fluctuation is much higher than for HEI-DSD and HEI-MMAP. For UCB-OK, the average



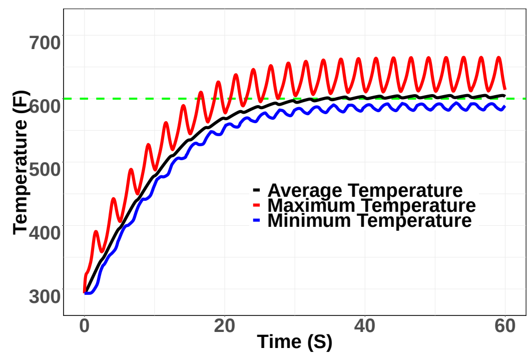

(a) Best objective values.



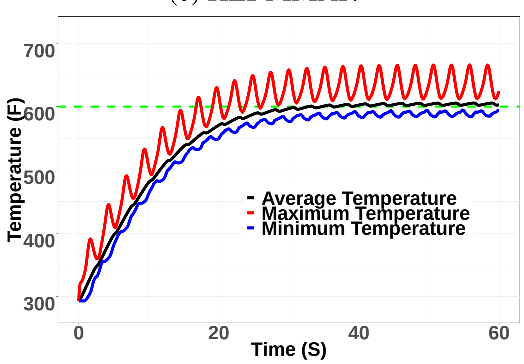

(b) SEI.



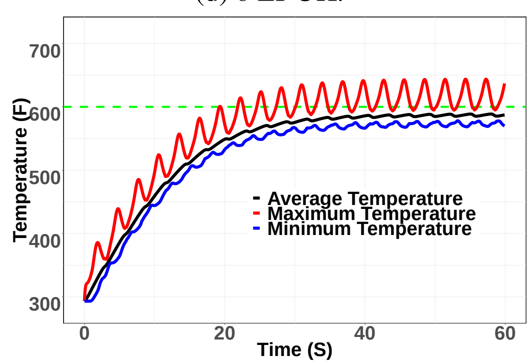

(c) HEI-MMAP.




(d) $\epsilon$-EI-UK.




(e) HEI-DSD.




(f) UCB-OK.

Figure 3.4: (a) shows the best objective value $f(\mathbf{x}_n^*)$ for the five compared methods. (b)-(f) show the average, maximum, and minimum temperature of the wafer over time, for each of the tested BO methods. The dotted green line marks the target temperature of $T^* = 600$ F.

temperature does not even reach the target temperature. The two proposed HEI methods (and $\epsilon$-EI-UK, although its performance is slightly worse) return noticeably improved settings compared to the two earlier methods, thereby providing engineers with an effective and robust wafer heating process for semiconductor manufacturing.

## 3.8 Conclusion

In this chapter, we presented a hierarchical expected improvement (HEI) framework for Bayesian optimization of a black-box objective $f$. HEI aims to correct a key limitation of the expected improvement (EI) method: its over-exploitation of the fitted GP model, which results in a lack of convergence to a global solution even for smooth objective functions. HEI addresses this via a hierarchical GP model, which integrates parameter uncertainty of the fitted model within a closed-form acquisition function. This provides a principled way for correcting over-exploitation by encouraging exploration of the optimization space. We then introduce several hyperparameter specification methods, which allow HEI to efficiently approximate a fully Bayesian optimization procedure. Under certain prior specifications, we prove the global convergence of HEI over a broad function class for $f$, and derive near-minimax convergence rates. In numerical experiments, HEI provides improved optimization performance over existing Bayesian optimization methods, for both simulations and a process optimization problem in semiconductor manufacturing.

# CHAPTER 4

# LEARNING TO DEFEND BY LEARNING TO ATTACK

## 4.1 Introduction

This decade has witnessed great breakthroughs in deep learning in a variety of applications, such as computer vision [100, 101, 102, 103]. Recent studies [104], however, show that most of these deep learning models are very vulnerable to adversarial attacks. Specifically, by injecting a small perturbation to a normal sample, one can obtain an adversarial sample. Although the adversarial sample is semantically indistinguishable from the normal one, it can fool deep learning models and undermine the security of deep learning, causing reliability problems in autonomous driving, biometric authentication, *etc*.

Researchers have devoted many efforts to study efficient adversarial attack and defense [104, 105, 106, 107, 108, 109]. There is a growing body of work on generating adversarial samples, *e.g.*, fast gradient sign method (FGSM, [105]), projected gradient method (PGM, [110]), Carlini-Wagner (CW, [111]), *etc*. As for defense, existing methods can be unified as a bilevel optimization problem as follows:

$$\text{(Leader)} \qquad \min_{\boldsymbol{\theta}} \ \ \mathbb{E}_{P^*}\left[\ell(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}),\tilde{\boldsymbol{y}})\right], \tag{4.1.1}$$

$$\text{(Follower)} \quad \text{subject to } \ P^* \in \operatorname{argmax}_{\tilde{P}\in\mathcal{P}} \mathbb{E}_{\tilde{P}}\left[q_{f_{\boldsymbol{\theta}}}\big((\boldsymbol{x},\boldsymbol{y}),(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\big)\right], \tag{4.1.2}$$

where $f_{\boldsymbol{\theta}}$ denotes the neural network classifier with parameter $\boldsymbol{\theta}$, $(\boldsymbol{x},\boldsymbol{y})$ denotes the clean sample from distribution $D$, $q_{f_{\boldsymbol{\theta}}}(\cdot,\cdot)$ denotes a measure depending on network $f_{\boldsymbol{\theta}}$, and $\mathcal{P}$ denotes a set of joint distributions of perturbed sample $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})$ and clean sample $(\boldsymbol{x},\boldsymbol{y})$. Here $\tilde{P}\in\mathcal{P}$ satisfies that in each sample $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})$ is close to $(\boldsymbol{x},\boldsymbol{y})$ and the marginal distribution of $\tilde{P}$ over $(\boldsymbol{x},\boldsymbol{y})$ is $D$. By solving (4.1.2), $P^*$ essentially represents an effective adversarial distribution. Existing adversarial training methods use different approaches to find $P^*$

under different $q_{f_{\boldsymbol{\theta}}}$ and $\mathcal{P}$. For example, [105] consider a special case of this problem, distributionally robust optimization (DRO, [112, 113]). In DRO, $q_{f_{\boldsymbol{\theta}}}$ in (4.1.2) is the same as $\ell$ in (4.1.1) and $\tilde{P} \in \mathcal{P}$ satisfies that in each sample $\tilde{\boldsymbol{y}} = \boldsymbol{y}$, *i.e.*, train the network $f_{\boldsymbol{\theta}}$ over adversarial samples and still require $f_{\boldsymbol{\theta}}$ to yield the correct labels. Another example is Adversarial Interpolation Training [114] where $q_{f_{\boldsymbol{\theta}}}$ is the cosine similarity between the features of adversarial sample and clean sample, and $\mathcal{P}$ is a set of adversarial distribution yielded by mixup [115]. More details are in Section 4.2.

In the optimization literature, (4.1.1) and (4.1.2) are referred to as leader and follower optimization problems, respectively. Such a bilevel formulation naturally provides us a unified perspective on prior works of robustifying the neural network: The leader aims to find a robust network with parameter $\boldsymbol{\theta}$ so that the loss given by the training distribution from the follower problem is minimized; The follower targets on finding an optimal distribution that maximizes a certain measure, which yields a distribution of adversarial samples.

Though the bilevel problem is straightforward and well formulated, it is hard to solve. Even the simplest version of bilevel problem, linear-linear bilevel optimization, is shown to be NP-hard [116]. In our case, the problem becomes even more challenging, since the loss function $\ell$ in the leader problem is highly nonconvex in $\boldsymbol{\theta}$ and the follower targets on finding an optimal distribution under a nonconcave measure $q_{f_{\boldsymbol{\theta}}}$. Besides, in general, the feasible domain of the follower problem is a space of continuous distributions; while, in practice, we have only finite samples to approximate the original problem. Such a gap makes the problem even more challenging to solve.

There are several approaches to solve the original problem (4.1.1) and (4.1.2). Under the DRO setting, [105] propose to use FGSM to solve the DRO. However, [110] then find that FGSM with true label suffers from a "label leaking" issue, which ruins adversarial training. [108] further suggest to find adversarial samples by PGM and obtain a better result than FGSM, since FGSM essentially is one iteration PGM; Alternatively, [114] propose to combine FGSM and mixup to yield an adversarial samples for both feature and label. All these methods need to find an adversarial $(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}_i)$ for each clean sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, thus the dimension of the overall search space for all samples is substantial, which makes the computation expensive. More recently, [117] propose to use the natural evolution strategy to learn an adversarial distribution over feature under the black-box setting, which is beyond the scope of this chapter.

To address the above challenges, we propose a new learning-to-learn (L2L) framework that provides a more principled and efficient way for solving adversarial training. Specifically, we parameterize the optimizer of the follower problem by a neural network denoted by $g_{\boldsymbol{\phi}}(\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}, \boldsymbol{y}))$, where $\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}, \boldsymbol{y})$ denotes the input of the optimizer $g_{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}$ is the parameter of the optimizer. We also call the optimizer as the attacker. Since the neural network is very powerful in function approximation, our parameterization ensures that $g_{\boldsymbol{\phi}}$ is able to yield strong adversarial samples. Under our framework, instead of directly solving (4.1.2), we update the parameter $\boldsymbol{\phi}$ of the optimizer $g_{\boldsymbol{\phi}}$. Our training procedure becomes updating the parameters of two neural networks, which is quite similar to generative adversarial network (GAN, [49]). The proposed L2L is a generic framework and can be extended to other bilevel optimization problems, *e.g.*, generative adversarial imitation learning, which is studied in Appendix D.5.

Different from the hand-designed methods that compute the adversarial perturbation $\boldsymbol{\delta}_i = \tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i$ for each individual sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ using gradients from backpropagation, our methods generate the perturbations for all samples through the shared optimizer $g_{\boldsymbol{\phi}}$. This enables the optimizer $g_{\boldsymbol{\phi}}$ to learn potential common structures of the perturbations. Therefore, our method is capable of yielding strong perturbations and accelerating the training process. Furthermore, the L2L framework is very flexible: we can either choose different input $\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}, \boldsymbol{y})$, or use different architecture. For example, we can include gradient information in $\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}, \boldsymbol{y})$ and use a recurrent neural network (RNN) to mimic multi-step gradient-type methods. Instead of simply computing the high order information with finite difference approximation or multiple gradients, by parameterizing the algorithm as a neural network, our proposed methods can capture this information in a much adaptive way [118]. Our experiments demonstrate that our proposed method not only outperforms existing adversarial training methods, *e.g.*, PGM training, but also enjoys the computational efficiency over CIFAR-10 and CIFAR-100 datasets [119].

The research on L2L has a long history [120, 121, 122, 123, 124, 125]. The basic idea is that the updating formula of complicated optimization algorithms is first modeled in a parametric form, and then the parameters are learned by some simple algorithms, *e.g.*, stochastic gradient algorithm. Among existing works, [124] propose a system allowing the output of backpropagation from one network to feed into an additional learning network, with both networks trained jointly; Based on this, [125] further show that the design of an optimization algorithm can be cast as a learning problem. Specifically, they use long short-term memory RNNs to model the algorithm and allow the RNNs to exploit structure in the problems of interest in an adaptive way, which is undoubtedly one of the most popular methods for learning-to-learn.

However, there are two major drawbacks of the existing L2L methods: **(1)** It requires a large amount of datasets (or a large number of tasks in multi-task learning) to guarantee the learned optimizer to generalize, which significantly limits their applicability (most of the related works only consider the image encoding as the motivating application); **(2)** The number of layers/iterations in RNNs for modeling algorithms cannot be large to avoid significant computational burden in backpropagation.

Our contribution is that we fill the blank of L2L framework in solving bilevel optimization problems, and our proposed methods do not suffer from the aforementioned drawbacks: **(1)** Different $f_{\boldsymbol{\theta}}$ and $(\boldsymbol{x}, \boldsymbol{y})$ essentially yield different follower problems. Therefore, for adversarial training, we have sufficiently many tasks for learning-to-learn; **(2)** The follower problem does not need a large scale RNN, and we use a convolutional neural network (CNN) or a length-two RNN (sequence of length equals 2) as our attacker network, which eases computation.

**Notations**. Given $a \in \mathbb{R}$, denote $(a)_+$ as $\max(a, 0)$. Given two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, denote $x_i$ as the $i$-th element of $\boldsymbol{x}$, $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$ as the $\ell_\infty$-norm of $\boldsymbol{x}$, $\boldsymbol{x} \circ \boldsymbol{y} = [x_1y_1, \cdots, x_dy_d]^\top$ as element-wise product, and $\boldsymbol{e}_i$ is the vector with $i$-th element as 1 and others as 0. Denote the simplex in $\mathbb{R}^d$ by $\Delta(d)$, the $\ell_\infty$-ball centered at $\boldsymbol{x}$ with radius $\epsilon$ by $\mathcal{B}(\boldsymbol{x}, \epsilon) = \{\boldsymbol{y} \in \mathbb{R}^d : \|\boldsymbol{y} - \boldsymbol{x}\|_\infty \leq \epsilon\}$ and the projection to $\mathcal{B}(0, \epsilon)$ as $\Pi_\epsilon(\boldsymbol{\delta}) = \operatorname{sign}(\boldsymbol{\delta}) \circ \max(|\boldsymbol{\delta}|, \epsilon)$, where sign and max are element-wise operators.

## 4.2 Preliminary

This chapter mainly focuses on the defense for $\ell_\infty$-norm attack. In this section, we first introduce two popular cases of the original problem in the literature: distributionally robust optimization (DRO) and adversarial interpolation training (AIT). Then we discuss the fundamental hardness of solving the original problem and the drawbacks of existing approaches.

### 4.2.1 Adversarial Training

Instead of using population loss in problem (4.1.1) and (4.1.2), we use empirical loss in the following context, since in practice we only have finite samples. Given $n$ samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, where $\boldsymbol{x}_i$ is the $i$-th image and $\boldsymbol{y}_i$ is the one-hot vector for the corresponding label, DRO solves the following problem:

$$\min_{\boldsymbol{\theta}} \ \frac{1}{n} \sum_{i=1}^{n} \left[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i), \boldsymbol{y}_i)\right], \tag{4.2.1}$$

$$\text{subject to} \quad \boldsymbol{\delta}_i \in \operatorname*{argmax}_{\boldsymbol{\delta} \in \mathcal{B}(0,\epsilon)} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\delta}), \boldsymbol{y}_i). \tag{4.2.2}$$

The standard pipeline of DRO version is shown in Algorithm 2. Since the step of generating adversarial perturbation $\boldsymbol{\delta}_i$ in Algorithm 2 is intractable, most adversarial training methods adopt hand-designed algorithms. For example, [110] propose to solve follower problem (4.2.2) approximately by first order methods such as PGM. Specifically, PGM iteratively updates the adversarial perturbation by the projected sign gradient ascent method for each sample: Given one sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, at the $t$-th iteration, PGM takes

$$\boldsymbol{\delta}_i^t \leftarrow \Pi_\epsilon\big(\boldsymbol{\delta}_i^{t-1} + \eta \cdot \operatorname{sign}\big(\nabla_{\boldsymbol{x}} \ell(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_i^t), \boldsymbol{y}_i)\big)\big), \tag{4.2.3}$$

where $\tilde{\boldsymbol{x}}_i^t = \boldsymbol{x}_i + \boldsymbol{\delta}_i^{t-1}$, $\eta$ is the perturbation step size, $T$ is a pre-defined total number of iterations, and $\boldsymbol{\delta}_i^0 = \mathbf{0}$, $t = 1, \cdots, T$. Finally PGM takes $\boldsymbol{\delta}_i = \boldsymbol{\delta}_i^T$. Note that FGSM essentially is one-iteration PGM. Besides, some works adopt other optimization methods, e.g., momentum gradient method [126], and L-BFGS [127].

**Algorithm 2** Distributionally Robust Optimization.

**Input:** $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$: data, $\alpha$: learning rate, $N$: number of iterations, $\epsilon$: perturbation magnitude.

**for** $t \leftarrow 1$ *to* $N$ **do**
  Sample a minibatch $\mathcal{M}_t$
  **for** $i$ *in* $\mathcal{M}_t$ **do**
    $\boldsymbol{\delta}_i \leftarrow \operatorname{argmax}_{\boldsymbol{\delta} \in \mathcal{B}(0,\epsilon)} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\delta}), \boldsymbol{y}_i)$ // Generate adversarial data.
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \frac{1}{|\mathcal{M}_t|} \sum_{i \in \mathcal{M}_t} \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i), \tilde{\boldsymbol{y}}_i)$ //Update $\boldsymbol{\theta}$ over adversarial data.

### 4.2.2 Adversarial Interpolation Training

Alternatively, AIT adopts the mixup method to generate an adversarial distribution for a given sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ and then randomly select a sample $(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}_i)$ from this adversarial distribution. Specifically, AIT solves the following problem:

$$\min_{\boldsymbol{\theta}} \ \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}_i) \sim D_i} \big[\ell(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_i), \tilde{\boldsymbol{y}}_i)\big], \tag{4.2.4}$$

where $D_i = \{(\tilde{\boldsymbol{x}}_i^j, \tilde{\boldsymbol{y}}_i^j)\}_{j=1}^n$ is generated as follows:

$$\tilde{\boldsymbol{x}}_i^j = \operatorname*{argmin}_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x}_i, \epsilon)} \frac{f_{\boldsymbol{\theta}}^s(\boldsymbol{x}_j) \cdot f_{\boldsymbol{\theta}}^s(\tilde{\boldsymbol{x}})}{\|f_{\boldsymbol{\theta}}^s(\boldsymbol{x}_j)\|_2 \|f_{\boldsymbol{\theta}}^s(\tilde{\boldsymbol{x}})\|_2}, \ \ \tilde{\boldsymbol{y}}_i^j = \operatorname*{argmin}_{\tilde{\boldsymbol{y}} \in \Delta(C) \cap \mathcal{B}(\boldsymbol{y}_i, \epsilon_{\boldsymbol{y}})} \|\tilde{\boldsymbol{y}} - \frac{\mathbf{1} - \boldsymbol{y}_j}{C-1}\|_2^2, \tag{4.2.5}$$

where $f^s_{\boldsymbol{\theta}}(\cdot)$ denotes the output of the $s$-th layer of network $f_{\boldsymbol{\theta}}$, and $C$ denotes the number of classes. The standard pipeline is shown in Algorithm 3. To ease the computation, [114] use one-step gradient update as the solution of (4.2.5).

---

**Algorithm 3** Adversarial Interpolation Training.

**Input:** $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$: data, $\alpha$: learning rate, $N$: number of iterations, $\epsilon, \epsilon_{\boldsymbol{y}}$: perturbation magnitudes, $s$: the output layer of network for the follower.

**for** $t \leftarrow 1$ *to* $N$ **do**
- Sample a minibatch $\mathcal{M}_t$
- **for** $i$ *in* $\mathcal{M}_t$ **do**
  - Sample another index $j$, $\tilde{\boldsymbol{y}}_i \leftarrow (1-\epsilon_y)\boldsymbol{y}_i + \epsilon_y(\mathbf{1}-\boldsymbol{y}_j)/(C-1)$,
  - $\tilde{\boldsymbol{x}}_i \leftarrow \mathrm{argmin}_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x}_i, \epsilon)} \frac{f^s_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \cdot f^s_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})}{\|f^s_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\|_2 \|f^s_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})\|_2}$ // Generate adversarial data.
- $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \frac{1}{|\mathcal{M}_t|} \sum_{i \in \mathcal{M}_t} \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_i), \tilde{\boldsymbol{y}}_i)$ // Update $\boldsymbol{\theta}$ over adversarial data.

---

### 4.2.3 Hardness

Ideally, we want to obtain the optima for the follower problem, *i.e.*,

$$P^* := \underset{\tilde{P} \in \mathcal{P}}{\mathrm{argmax}} \, \underset{\tilde{P}}{\mathbb{E}} \left[ q_{f_{\boldsymbol{\theta}}}\big((\boldsymbol{x}, \boldsymbol{y}), (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})\big) \right].$$

However, the measure $q_{f_{\boldsymbol{\theta}}}$ depends on network $f_{\boldsymbol{\theta}}$, which makes solving $P^*$ intractable. Therefore, in reality the sample $(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}_i)$ from the obtained solution $\tilde{P}$ is very unlikely to be the sample $(\boldsymbol{x}_i^*, \boldsymbol{y}_i^*)$ from $P^*$. This then often leads to a highly unreliable or even completely wrong search direction, *i.e.*,

$$\langle \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_i), \tilde{\boldsymbol{y}}_i), \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i^*), \boldsymbol{y}_i^*) \rangle < 0,$$

which may further results in a limiting cycle shown in Figure 4.1 (Detailed discussion is in Appendix D.1). This becomes even worse when sample noises exist. Moreover, among the methods mentioned earlier, except FGSM, all require numerous queries for gradients, which is computationally expensive.

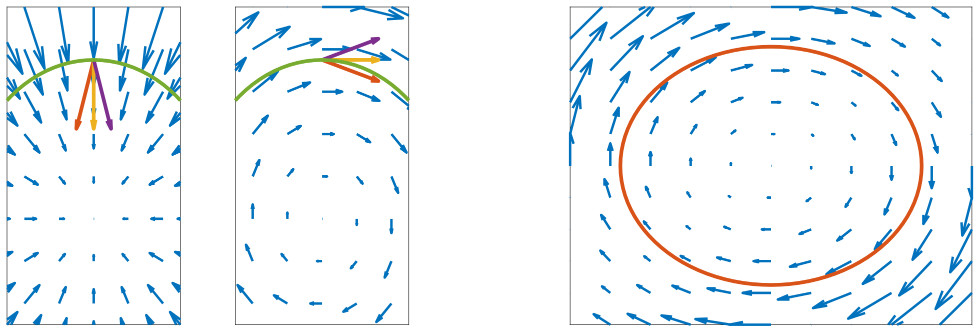



Figure 4.1: Illustration for the hardness of problem (4.1.1) and (4.1.2). A wrong update direction leads to a limiting cycle and algorithms fail to converge. More details in Appendix D.1.

### 4.3 Learning to Defense by Learning to Attack (L2L)

Since the hand-designed methods for bilevel problem (4.1.1) and (4.1.2) do not perform well, we propose to learn an optimizer for the follower problem. Specifically, we parameterize $\boldsymbol{\delta} = \tilde{\boldsymbol{x}} - \boldsymbol{x}$, the difference between the adversarial sample and clean input[1], by a neural network $g_{\boldsymbol{\phi}}(\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}, \boldsymbol{y}))$ with input $\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}, \boldsymbol{y})$ summarizing the information of data and classifier $f_{\boldsymbol{\theta}}(\cdot)$. We first show how our method works on the DRO approach: We convert DRO problem (4.2.1) and (4.2.2) to

$$\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + g_{\boldsymbol{\phi}}(\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i, \boldsymbol{y}_i))), \boldsymbol{y}_i), \tag{4.3.1}$$

where $\boldsymbol{\phi}^*$ is defined as the solution to the following optimization problem:

$$\boldsymbol{\phi}^* \in \operatorname*{argmax}_{\boldsymbol{\phi}} \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + g_{\boldsymbol{\phi}}(\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i, \boldsymbol{y}_i))), \boldsymbol{y}_i),$$

$$\text{subject to} \quad g_{\boldsymbol{\phi}}(\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i, \boldsymbol{y}_i)) \in \mathcal{B}(0, \epsilon), i \in [1, ..., n].$$

[1]This helps to handle the constraints $\boldsymbol{\delta} \in \mathcal{B}(0, \epsilon)$.

The optimizer $g_{\phi}$ targets on generating optimal perturbations under constraints $g_{\phi}(\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i, \boldsymbol{y}_i)) \in \mathcal{B}(0, \epsilon)$. These constraints can be easily handled by a $\tanh$ activation function and a $\epsilon$ scaler in the last layer of $g_{\phi}$.

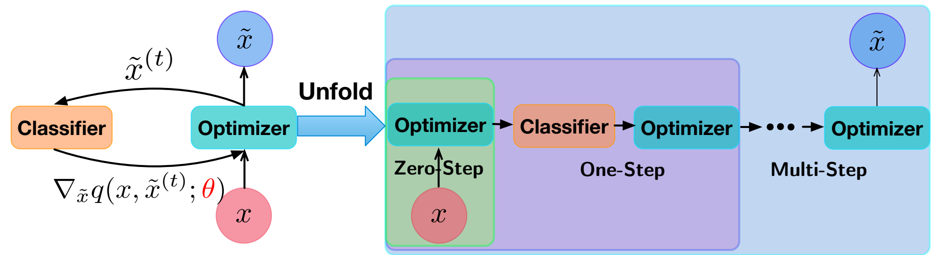



Figure 4.2: An illustration of L2L: A neural network models optimizer for generating attack.

This L2L framework is very flexible: We can choose different $\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}, \boldsymbol{y})$ as the input and mimic multi-step algorithms shown in Figure 4.2. Here we provide three examples for the DRO:

**Naive Attacker.** This is the simplest example among our methods, taking the original image $\boldsymbol{x}_i$ as the input, *i.e.*,

$$\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \boldsymbol{x}_i \quad \text{and} \quad \boldsymbol{\delta}_i = g_{\phi}(\boldsymbol{x}_i).$$

Under this setting, L2L training is similar to GAN training. The major difference is that the generator in GAN yields synthetic data by transforming random noises, while the naive attacker generates perturbations via training samples.

**Gradient Attacker.** Motivated by FGSM, we design an attacker which takes the gradient information into computation. Specifically, we concatenate the image $\boldsymbol{x}_i$ and the gradient $\nabla_{\boldsymbol{x}} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i)$ as the input of $g$, *i.e.*,

$$\mathcal{A}_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \big[\boldsymbol{x}_i, \nabla_{\boldsymbol{x}} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i)\big] \text{ and } \boldsymbol{\delta}_i = g_{\phi}\big([\boldsymbol{x}_i, \nabla_{\boldsymbol{x}} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i)]\big).$$

Since more information is provided, we expect the attacker network to be more effective to

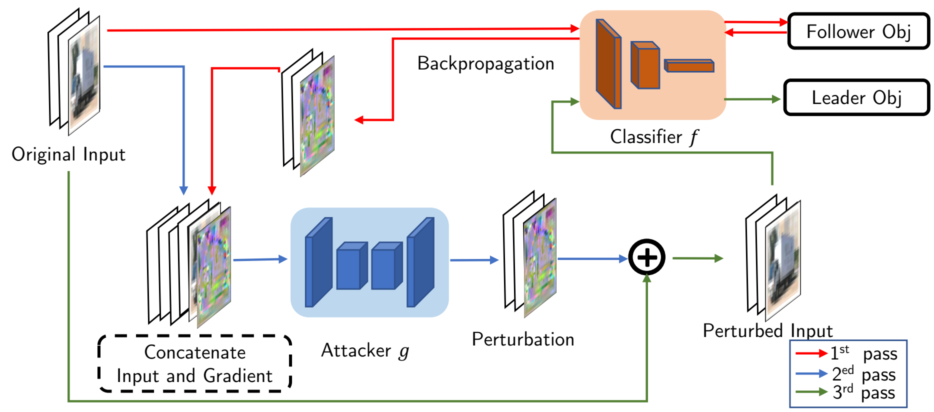



Figure 4.3: The architecture of PGM adversarial training with gradient attacker network.
learn and yield more powerful perturbations.

**Multi-Step Gradient Attacker.** Motivated by PGM, we adapt the RNN to mimic a multi-step gradient update. Specifically, we use the gradient optimizer network as the cell of RNN sharing the same parameter $\phi$. As we mentioned earlier, the number of layers/iterations in the RNN for modeling algorithms cannot be very large so as to avoid significant computational burden in backpropagation. In this chapter, we focus on a length-two RNN to mimic a two-step gradient update. The corresponding perturbation becomes:

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \Pi_\epsilon\big(\boldsymbol{\delta}_i^{(0)} + g_{\boldsymbol{\phi}}\big([\tilde{\boldsymbol{x}}_i^{(0)}, \nabla_{\boldsymbol{x}}\ell(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_i^{(0)}), \boldsymbol{y}_i)]\big)\big),$$

where $\tilde{\boldsymbol{x}}_i^{(0)} = \boldsymbol{x}_i + \boldsymbol{\delta}_i^{(0)}$ and $\boldsymbol{\delta}_i^{(0)} = g_{\boldsymbol{\phi}}\big([\boldsymbol{x}_i, \nabla_{\boldsymbol{x}}\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i)]\big)$.

**Algorithm 4** Learning-to-learn-based DRO with gradient attacker

**Input:** $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$: clean data, $\alpha_1, \alpha_2$: learning rates, $N$: number of epochs.
**for** $t \leftarrow 1$ *to* $N$ **do**
  Sample a minibatch $\mathcal{M}_t$
  **for** $i$ *in* $\mathcal{M}_t$ **do**
    $\boldsymbol{u}_i \leftarrow \nabla_{\boldsymbol{x}}\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i), \boldsymbol{\delta}_i \leftarrow g_{\boldsymbol{\phi}}([\boldsymbol{x}_i, \boldsymbol{u}_i])$ //Generate perturbation by $g_{\boldsymbol{\phi}}$.
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \frac{\alpha_1}{|\mathcal{M}_t|}\sum_{i\in\mathcal{M}_t}\nabla_{\boldsymbol{\theta}}\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i), \boldsymbol{y}_i)$ // Update $\boldsymbol{\theta}$ over adversarial data.
  $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \frac{\alpha_2}{|\mathcal{M}_t|}\sum_{i\in\mathcal{M}_t}\nabla_{\boldsymbol{\phi}}\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i), \boldsymbol{y}_i)$ // Update $\boldsymbol{\phi}$ over adversarial data.

Taking gradient attackers as an example, Figure 4.3 illustrates how L2L works and jointly trains two networks: The first forward pass is used to obtain gradient of the classification loss over the clean data; The second forward pass is used to generate perturbation $\boldsymbol{\delta}_i$ by the attacker $g$; The third forward pass is used to calculate the adversarial loss $\ell$ in (4.3.1). Since our gradient attacker network only needs one backpropagation to query gradient, it amortizes the adversarial training cost, which leads to better computational efficiency. Moreover, L2L may adapt to the underlying optimization problem and yield better solution for the follower problem. The corresponding procedure of L2L is shown in Algorithm 4.

---

**Algorithm 5** Learning-to-learn with Adversarial Interpolation Training

---

**Input:** $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$: data, $\alpha$: learning rate, $N$: number of iterations, $\epsilon, \epsilon_{\boldsymbol{y}}$: perturbation magnitudes.

**for** $t \leftarrow 1$ *to* $N$ **do**

  Sample a minibatch $\mathcal{M}_t$

  **for** $i$ *in* $\mathcal{M}_t$ **do**

    Sample another index $j$, $\tilde{\boldsymbol{y}}_i \leftarrow (1-\epsilon_{\boldsymbol{y}})\boldsymbol{y}_i + \epsilon_{\boldsymbol{y}}(\mathbf{1}-\boldsymbol{y}_j)/(C-1)$,

    $\boldsymbol{u}_i = \nabla_{\boldsymbol{x}_i} q_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{\delta}_i \leftarrow g_{\boldsymbol{\phi}}(\boldsymbol{x}_i, \boldsymbol{u}_i)$ //Generate perturbation by $g_{\boldsymbol{\phi}}$.

  $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \frac{\alpha_2}{|\mathcal{M}_t|} \sum_{i \in \mathcal{M}_t} \nabla_{\boldsymbol{\phi}} q_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j)$ //Update $\boldsymbol{\phi}$ over adversarial data.

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \frac{\alpha_1}{|\mathcal{M}_t|} \sum_{i \in \mathcal{M}_t} \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i), \tilde{\boldsymbol{y}}_i)$ //Update $\boldsymbol{\theta}$ over adversarial data.

---

It is straightforward to extend L2L to AIT as shown in Algorithm 5. For the feature perturbation, we simply replace the gradient of $\ell$, $\nabla_{\boldsymbol{x}} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i)$, by the gradient of $q_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{f^s_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \cdot f^s_{\boldsymbol{\theta}}(\boldsymbol{x}_j)}{\|f^s_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\|_2 \|f^s_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\|_2}$, $\nabla_{\boldsymbol{x}_i} q_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in the attacker input. Taking gradient network as an example, given a sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, we first randomly select another sample $(\boldsymbol{x}_j, \boldsymbol{y}_j)$, and yield the adversarial training feature as follows:

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + g_{\boldsymbol{\phi}}\Big([\boldsymbol{x}_i, \nabla_{\boldsymbol{x}_i} q_{f_{\boldsymbol{\theta}}}(\boldsymbol{x}_i, \boldsymbol{x}_j)]\Big), \tag{4.3.2}$$

and adopt the corresponding label vector $\tilde{\boldsymbol{y}}_i$ from (4.2.5).

## 4.4 Experiments

To demonstrate the effectiveness and computational efficiency of our methods, we conduct experiments over both CIFAR-10 and CIFAR-100 datasets [119]. We compare our methods with original PGM training and adversarial interpolation training. All implementation are done in PyTorch with one single NVIDIA 2080 Ti GPU. Here we discuss the white-box setting, which is the most direct way to evaluate the robustness.

**Classifier Network.** All experiments adopt a 34-layer wide residual network (WRN-34-10, [128]) implemented by [129] as the classifier network. For each method, we train the classifier network from scratch.

Table 4.1: Attacker architecture: $k, c, s, p$ denote the kernel size, output channels, stride and padding parameters of convolutional layers, respectively.

| | |
|---|---|
| Conv: | $[k = 3 \times 3, c = 64, s = 1, p = 1]$, BN+ReLU |
| ResBlock: | $[k = 3 \times 3, c = 128, s = 1, p = 1]$ |
| ResBlock: | $[k = 3 \times 3, c = 256, s = 1, p = 1]$ |
| ResBlock: | $[k = 3 \times 3, c = 128, s = 1, p = 1]$ |
| Conv: | $[k = 3 \times 3, c = 3, s = 1, p = 1]$, $\tanh$ |

**Attacker.** Table 4.1 presents the architecture of our attacker network[2]. The ResBlock uses the same structure as the generator proposed in [130]. The detailed structure of ResBlock is provided in Appendix D.3. Batch normalization (BN) and activations, *e.g.*, ReLU and $\tanh$, are applied when specified. The $\tanh$ function can easily make the output of attacker satisfy the constraints.

**White-box and Black-box[3].** We compare different methods under both white-box and black-box settings. Under the white-box setting, attackers can access all parameters of target models and generate adversarial examples based on the models; whereas under the black-box setting, accessing parameters is prohibited. Therefore, we adopt the standard transfer attack method from [131].

**Robust Evaluation[4].** We evaluate the robustness of the networks by PGM and CW attacks with the maximum perturbation magnitude $\epsilon = 0.031$ (after rescaling the pixels to $[0, 1]$)

[2]We provide another attacker architecture with down-sampling modules in the Appendix D.3. With such an attacker, L2L adversarial training is less stable, but faster.

[3]Due to the space limit, we leave the results of the black-box setting in Appendix D.2.

[4]More detailed robustness checklist is provided in Appendix D.4.

over CIFAR-10 and CIFAR-100. For PGM attack, we use 20 and 100-iteration PGM with a perturbation step size $\eta = 0.003$, and for each sample we initialize the adversary perturbation randomly in the $\mathcal{B}(0, 10^{-4})$. For CW attack, we adopt the implementation from [111], and set the maximum number of iterations as $100$. For each method, we repeat $5$ runs with different random initial seed and report the *worst result*. For CIFAR-10, we also evaluate the robustness of our Grad L2L and 2-Step L2L networks using random attacks, for which we uniformly sample $10^5$ perturbations in $\mathcal{B}(0, 0.031)$ adding to each test sample. We also evaluate the robustness of our Grad L2L and 2-Step L2L networks under their own attackers.

### 4.4.1 PGM Training

For simplicity, we denote PGM Net as the classifier with PGM training, and Naive L2L, Grad L2L, and 2-Step L2L as the classifiers using L2L training with corresponding attackers.

**Original PGM.** For CIFAR-10, we directly report the result from [108] as the baseline; For CIFAR-100, we train a PGM Net as the baseline: To update the classifier's parameter $\boldsymbol{\theta}$, we use the stochastic gradient descent (SGD) algorithm with Polyak's momentum (parameter $0.9$, [132]) and weight decay (parameter $2 \times 10^{-4}$, [133]). In addition, we adapt the setting from [108] but train the network for $100$ epochs with initial learning rate $0.1$, decay schedule [30,60,90], and decay rate $0.1$. We use a 10-iteration PGM with the perturbation step size $0.007$ in (4.2.3) to generate adversarial samples.

**PGM+L2L.** We train two networks for 100 epochs. To update classifier's parameter $\boldsymbol{\theta}$, we use the same configuration as original PGM training; To update the attacker's parameter $\boldsymbol{\phi}$, we use Adam optimizer (parameter $[0.9, 0.999]$, [134]) with initial learning rate $10^{-3}$ (no learning rate decay) and weight decay (parameter $2 \times 10^{-4}$) so that it adaptively balances the updates in both leader and follower optimization problems.

**Experiment Results.** Table 4.2 shows the results of all PGM training methods over CIFAR-

Table 4.2: Results of different defense methods under the white-box setting.

| Defense Method | Attack | Dataset | Clean Accuracy | Robust Accuracy |
|---|---|---|---|---|
| Stability Training [107] | PGM-20 | CIFAR10 | 94.64% | 0.15% |
| PGM Net [108] | PGM-20 | CIFAR10 | 87.30% | 47.04% |
| Naive L2L | PGM-20 | CIFAR10 | 94.53% | 0.01% |
| Grad L2L | PGM-20 | CIFAR10 | 85.84% | 51.17% |
| 2-Step L2L | PGM-20 | CIFAR10 | 85.35% | 54.32% |
| Grad L2L | PGM-100 | CIFAR10 | 85.84% | 47.72% |
| 2-Step L2L | PGM-100 | CIFAR10 | 85.35% | 52.12% |
| Grad L2L | CW | CIFAR10 | 85.84% | 53.5% |
| 2-Step L2L | CW | CIFAR10 | 85.35% | 57.07% |
| Grad L2L | Random | CIFAR10 | 85.84% | 82.67% |
| 2-Step L2L | Random | CIFAR10 | 85.35% | 83.10% |
| Grad L2L | Grad L2L | CIFAR10 | 85.84% | 49.68% |
| 2-Step L2L | 2-Step L2L | CIFAR10 | 85.35% | 52.71% |
| PGM Net | PGM-20 | CIFAR100 | 62.68% | 23.75% |
| Grad L2L | PGM-20 | CIFAR100 | 62.18% | 28.67% |
| 2-Step L2L | PGM-20 | CIFAR100 | 60.95% | 31.03% |
| PGM Net | PGM-100 | CIFAR100 | 62.68% | 22.06% |
| Grad L2L | PGM-100 | CIFAR100 | 62.18% | 26.69% |
| 2-Step L2L | PGM-100 | CIFAR100 | 60.95% | 29.75% |
| PGM Net | CW | CIFAR100 | 62.68% | 25.95% |
| Grad L2L | CW | CIFAR100 | 62.18% | 29.65% |
| 2-Step L2L | CW | CIFAR100 | 60.95% | 32.28% |

10 and CIFAR-100 under the white-box setting. As can be seen, without gradient information, Naive L2L is vulnerable to the PGM attack. However, when the attacker utilizes the gradient information, Grad L2L and 2-Step L2L *significantly outperform* the PGM Net over CIFAR-10 and CIFAR-100, with a slight loss for the clean accuracy. From the experiments on CIFAR-10, our Grad L2L and 2-Step L2L are robust to random attacks, where the accuracy is only slightly lower than the clean accuracy. Furthermore, the accuracy of our Grad/2-Step L2L model under the Grad/2-Step L2L attacker is comparable to the accuracy under PGM attacks, which shows that L2L attackers are able to generate strong attacks. As can be seen, PGM-100 is stronger than Grad L2L attacker ($47.72\%$ vs. $49.68\%$), but similar to the 2-Step L2L attacker ($52.07\%$ vs. $52.71\%$). This means 2-Step L2L attacker is much stronger than Grad L2L attacker and explains why 2-Step L2L is stronger than Grad L2L and PGM net.

In addition, Table 4.3 shows the one epoch running time of all methods over CIFAR-10 and CIFAR-100. As can be seen, Grad L2L and 2-Step L2L is much faster than PGM Net.

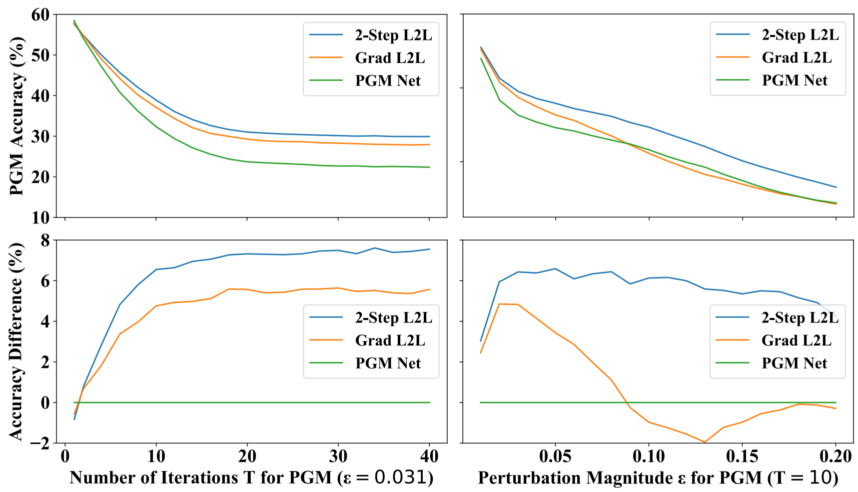



Figure 4.4: Robust accuracy against perturbation magnitude and number of iteration of PGM over CIFAR-100 adversarial samples;. (Top) Absolute accuracy; (Bottom) Performance gain over PGM Net. More results are provided in Appendix D.4.

By further comparing the accuracy of Grad/2-Step L2L and PGM Net in Table 2, we find that our proposed L2L methods enjoy computational efficiency. In addition, Figure 4.4 presents the robust accuracy against number of iterations (fixed perturbation magnitude $\epsilon = 0.031$) and perturbation magnitude (fixed number of iterations $T = 10$). As can be seen, 2-Step L2L is much more robust than PGM Net.

Table 4.3: One epoch running time. (Unit: s)

| Dataset | Plain Net | PGM Net | Naive L2L | Grad L2L | 2-Step L2L |
|---|---|---|---|---|---|
| CIFAR-10 | $106.5 \pm 1.5$ | $1310.8 \pm 14.2$ | $293.7 \pm 3.1$ | $617.5 \pm 6.1$ | $805.1 \pm 8.1$ |
| CIFAR-100 | $106.9 \pm 1.4$ | $1354.8 \pm 14.1$ | $310.0 \pm 2.9$ | $623.1 \pm 6.3$ | $824.7 \pm 8.4$ |

### 4.4.2 Adversarial Interpolation Training

We conduct the experiments of AIT over CIFAR-10 using the code from [114]. [5]

**Original AIT.** We follow the experimental setting in [114], but use a WRN-34-10. To update classifier's parameter $\boldsymbol{\theta}$, we use the same configuration in original PGM training. We choose the perturbation magnitude over label $\epsilon_{\boldsymbol{y}}$ as $0.5$. In addition, we train the whole network for 200 epochs with initial learning rate $0.1$, decay schedule [60,90], and decay

[5] `https://github.com/Adv-Interp/adv_interp`

rate $0.1$. Moreover, in each epoch, we first use FGSM to yield training samples via (4.2.5), and then train the AIT Net over these adversarial samples.

**AIT+L2L.** We train two networks for 200 epochs. To update classifier's parameter $\boldsymbol{\theta}$, we adopt the configuration of SGD from the original AIT; To update attacker's parameter $\phi$, we use Adam optimizer (parameter $[0.9, 0.999]$) with initial learning rate as $10^{-3}$ (no learning rate decay) and weight decay (parameter $2 \times 10^{-4}$).

Table 4.4: Results of AIT based defense methods under the white-box setting.

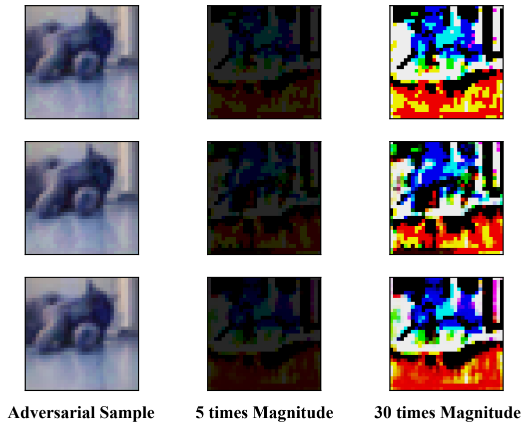
| Defense Method | Attack | Dataset | Clean Accuracy | Robust Accuracy |
|---|---|---|---|---|
| AIT | PGM-20 | CIFAR10 | 90.43% | 75.33% |
| Grad L2L | PGM-20 | CIFAR10 | 91.65% | 80.87% |
| AIT | PGM-100 | CIFAR10 | 90.43% | 67.84% |
| Grad L2L | PGM-100 | CIFAR10 | 91.65% | 79.20% |
| AIT | CW-20 | CIFAR10 | 90.43% | 64.79% |
| Grad L2L | CW-20 | CIFAR10 | 91.65% | 74.88% |
| AIT | CW-100 | CIFAR10 | 90.43% | 61.69% |
| Grad L2L | CW-100 | CIFAR10 | 91.65% | 73.46% |

**Experiment Results.** Table 4.4 shows the results of AIT methods over CIFAR-10 under the white-box setting. As can be seen, Grad L2L *significantly improves upon* the AIT Net over CIFAR-10 on both clean accuracy and robust accuracy.

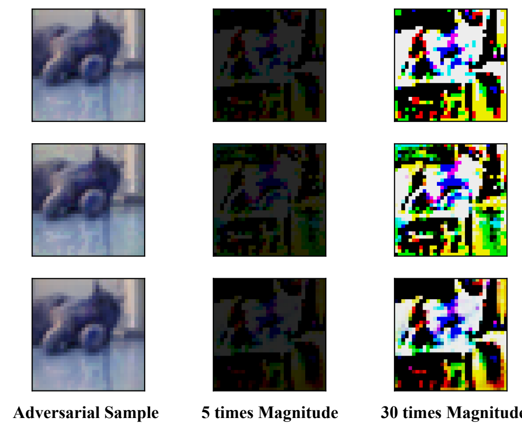
### 4.4.3 Visualization of Adversarial Examples

Figure 4.5 provides an illustrative example of adversarial perturbations generated by FGSM, PGM-20 and 2-Step L2L attacker for a *cat* in CIFAR-10. As can be seen, attacks for these two networks are very different. Moreover, the perturbation generated by the 2-Step L2L attacker is much more smooth than FGSM and PGM. In this example, 2-Step L2L labels all adversarial samples correctly; whereas the PGM Net is fooled by PGM-20 attack and misclassifies it as a *dog*.

Figure 4.6 provides an illustrative example of adversarial perturbations generated by PGM, AIT and Grad L2L for a *dog* in CIFAR-10. As can be seen, attacks for these two networks are very different: the attacks for the Grad L2L is more abundant in three channels. In this example, Grad L2L labels all adversarial samples correctly; whereas the AIT is fooled by all attacks and misclassifies it as a *horse*.

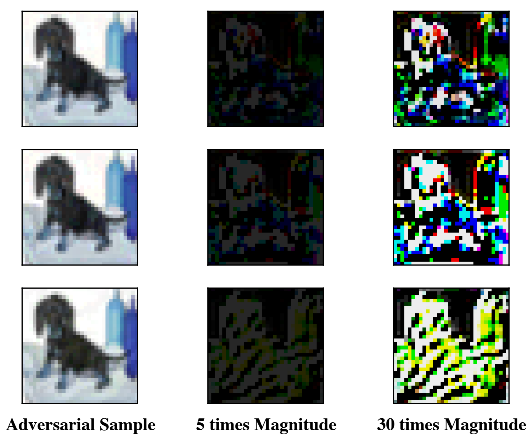



(a) PGM Net adversarial samples.

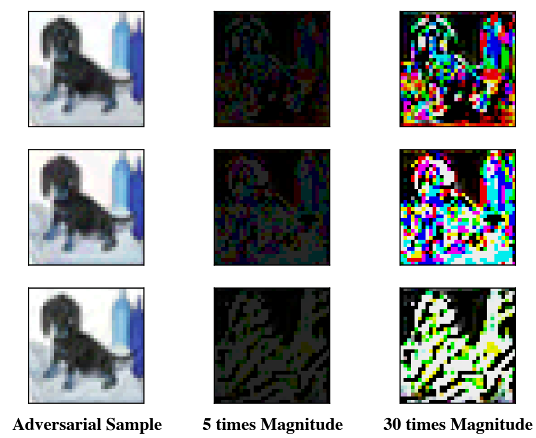



(b) 2-Step L2L adversarial samples.

Figure 4.5: Illustrative adversarial examples of FGSM (Top), PGM-20 (Mid), and 2-Step L2L (Bottom) perturbations for a cat under PGM Net and 2-Step L2L with $\epsilon = 0.031$.




(a) AIT adversarial samples.




(b) Grad L2L adversarial samples.

Figure 4.6: Illustrative adversarial examples of PGM-20 (Top), AIT (Mid), and Grad L2L (Bottom) perturbations for a dog under AIT Net and Grad L2L with $\epsilon = 0.031$.

## 4.5 Discussions

We discuss several closely related works:

• By leveraging the Fenchel duality and feature embedding technique, [135] convert a learning conditional distribution problem to a minimax problem, which is similar to our naive attacker. Both approaches, however, lack the primal information. In contrast, gradient attacker network considers the gradient information of primal variables, and achieves good results.

• [49] propose the GAN, which is very similar to our L2L framework. Both GAN and L2L contain one generator network and one classifier network, and jointly train these two networks. There are two major difference between GAN and our framework: (1) GAN aims to transform the random noises to the synthetic data which is similar to the training examples, while ours targets on transforming the training examples to the adversarial examples for robustifying the classifier; (2) Our generator network does not only take the training examples (analogous to the random noise in GAN) as the input, but also exploits the gradient information of the objective function, since it essentially represents an optimization algorithm. The training procedure of these two, however, are quite similar. We adopt some tricks from GAN training to our framework to stabilize training process, *e.g.*, in Grad L2L, we use the two-time scale trick [136].

• There are some other works simply combining the GAN framework and adversarial training together. For example, [137] and [138] propose some ad hoc GAN-based methods to robustify neural networks. Specifically, for generating adversarial examples, they only take training examples as the input of the generator, which lacks the information of the outer mimnimization problem. Instead, our proposed L2L methods (*e.g.*, Grad L2L, 2-step L2L) connect outer and inner problems by delivering the gradient information of the objective function to the generator. This is a very important reason for our performance gain on the benchmark datasets. As a result, the aforementioned GAN-based methods are only robust to simple attacks, *e.g.*, FGSM, on simple data sets, *e.g.*, MNIST, but fail for strong attacks, *e.g.*, PGM and CW, on complicated data sets, *e.g.* CIFAR, where our L2L methods achieve significantly better performance.

**Training Stability**: For improving the training stability, we use both clean image and the corresponding gradient as the input of the attacker. Without such gradient information, the attacker severely suffers from training instability, *e.g.*, the Naive Attacker Network. Furthermore, we try another architecture with downsampling modules, called "slim attacker" in Appendix D.3. We observed that the slim attacker also suffers from training instability. We suspect that the downsampling causes the loss of information. Thus, we tried to enhance the slim attacker by skip layer connections. In this way, the training is stabilized. However, the robust performance is still worse than the proposed architecture.

**Benefits of our L2L approach in adversarial training**:

**(1)** Since the neural network has been known to be powerful in function approximation, our attacker network $g$ can yield strong adversarial perturbations. Since they are generated by the same attacker, the attacker $g$ essentially learns some common structures across all samples;

**(2)** *Overparametrization* is conjectured to ease the training of deep neural networks. We believe that similar phenomena happen to our attacker network, and ease the adversarial training.

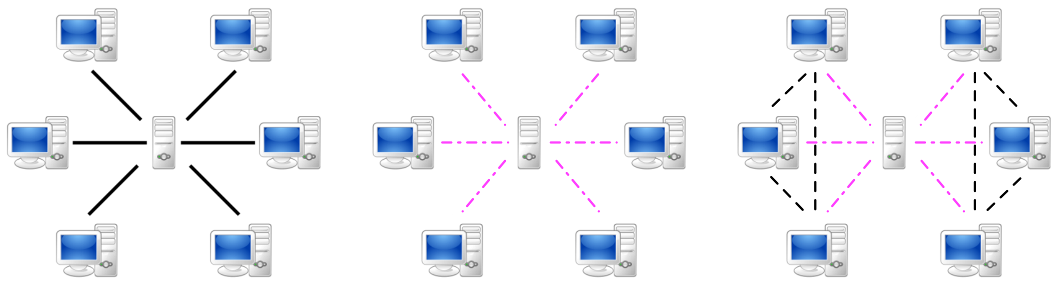
**(3)** Our proposed L2L framework is well structured and can be generalized to solve more complicated bilevel problems, *e.g.*, GAIL (see Appendix D.5). Taking our results as a start, we expect more principled and stronger follow-up work that applies L2L to solve the bilevel problems.

# CHAPTER 5
# CONDITIONAL AUTOREGRESSIVE DETECTION (CARD) FOR DISTRIBUTED NETWORK MONITORING

## 5.1 Introdution

With recent exciting developments in distributed computing technology, the problem of distributed network monitoring has become increasingly important in a broad range of applications, ranging from cybersecurity [139], environmental monitoring [140], social network monitoring [141, 142], and neuroscience [143]. In many such applications, large delays in detecting changes (or failure to detect changes) over the network may incur significant costs and damages. For example, a major Turkish oil pipeline failure resulted from a failure to detect signal changes [144]. This motivates the need for network monitoring methods, which not only leverage spatial information for efficient detection, but also do so in a cost-efficient, distributed manner.

In the change-point detection literature, various methods have been developed for network detection, based on the classic cumulative sum (CUSUM, [145]) and Shiryaev-Roberts [146, 147] statistics under independent assumptions [148, 149, 150]. Recently, researchers further exploited the spatial-temporal structures of the network data [151, 152, 153, 154, 155] for improving the detection power. However, these methods require the monitoring system to have a *fusion center* that gathers information, *i.e.*, raw data or statistics, from all sensors to perform decisions globally shown in Figure 5.1a. There are several limitations for these centralized methods with the modern network data: (1) limited communication bandwidth: in distributed geophysical sensor networks [140], sensors can only communicate with their neighboring sensors, but cannot communicate to far-away sensors or fusion center with too much data, since the channel bandwidth is interference limited; (2) communication delay: for seismic early warning systems, it is also not ideal for seismic sensors to send all information to a fusion hub and perform a global decision, but rather let them to make local decision, to avoid two-way communication delay; (3) computational limits: the network scale is quite large in modern applications, and it is computationally expensive or infeasible for getting a monitor statistics with the whole network data.



(a) Centralized Framework (b) One-shot Framework (c) Decentralized Framework

Figure 5.1: An illustration for the distributed change-point detection frameworks. Black solid line denotes the communication with raw data or statistics, dash line denotes the communication with statistics, and the pink dot dash line denotes the communication with one-bit decision.

To overcome these limitations, [156, 157] proposed a *one-shot* framework, which essentially uses the *marginal distribution* of the data on each sensor to monitor the whole network. Specifically, the system considers the observations as node-wise independent data, and each sensor makes a local decision with its own data. Once a sensor raises a local alarm, it transmits a one-bit signal to the fusion center shown in Figure 5.1b. Moreover, [156] established the theoretical guarantees for the one-shot method under node-wise independent assumptions. In practice, however, not only is such an assumption unrealistic, but also leads to significant reduction in detection power for the monitoring procedure due to ignoring the spatial correlation among the data.

For designing a scalable detection mechanism that utilizes the neighbor information among the modern network, it is important to perform a decentralized detection. Instead of having to send all information or statistics to a fusion center to form a global decision or making a local decision with the data on a single sensor, sensors perform local decisions with their neighbors' information. As shown in Figure 5.1c, each sensor communicates with its own neighbors and then makes a local decision. Recently, [158] proposed a decentralized consensus detection method, *i.e.*, make decisions locally and improve the detection power by consensus among the local community (the node and its neighbor nodes). They, however, further assumed that the data on all sensors are not only independent but also identically distributed, which is quite stringent in practice.

To break these stringent assumptions and leverage the spatial correlation, we propose a model-based approach that uses a Conditional AutoRegressive (CAR, [159]) model to capture the spatial correlation among the network data, and develop a new decentralized online change-point detection method, called Conditional AutoRegressive Detection (CARD). The key idea is the *conditional specification* for a CAR model, that is the joint distribution of the whole network observations is equivalent to a list of local conditional distributions (see Brook's lemma [160]). This further allows us to leverage the spatial correlation and monitor the whole network system in a distributed manner. Furthermore, to show the advantage of the proposed CARD method, we compare CARD and the one-shot method under the widely used metrics in sequential testing, average running length (ARL) and expected detection delay (EDD) [161], which are analogous to type I and type II errors in the classical hypothesis testing [162]. We then establish the ARL and EDD for both methods under the CAR model, and then show that CARD outperforms the one-shot method by leveraging the spatial correlation. Then we discuss how the network structure and the correlation affect the improvement of CARD over the one-shot method. We then conduct some empirical experiments in a suite of numerical simulations and two applications: power grid monitoring, and sparse population coding of biological neural networks, which also supports our theoretical results. Moreover, in the sparse coding application, CARD can find all *root causes* while the one-shot method misses one.

The rest of the paper is organized as follows: Section 5.2 formulates the problem, reviews the one-shot method, and establishes the corresponding ARL and EDD. Section 5.3

first introduces the CAR model and CARD method, then establishes the corresponding ARL and EDD, and compares CARD and the one-shot method. Section 5.4 then provides methodological developments on parameter estimates and the detection algorithm. Sections 5.5 and 5.6 compare CARD and the one-shot method for synthetic data generated with the simulated network, as well as two applications, power network monitoring, and sparse population coding of biological neural networks. Concluding remarks are given in Section 5.7.

## 5.2 Background and Motivation

We first describe the setup for the sensor network detection which is widely used in the power network monitoring, and then review some existing approaches and their drawbacks, which motivates our proposed method.

**Sensor Network Intrusion.** We represent a sensor network by an undirected connected graph $G = \{V, E\}$, where $V = \{v_1, ...v_p\}$ denotes the sensor set and $E$ denotes the edge set. Two sensors can communicate with each other if and only if there is an edge between these two sensors. Then we use $N(i)$ to denote the neighbor of sensor $i$, *i.e.*, $N(i) = \{j : v_j \in V, (v_i, v_j) \in E\}$. Without loss of generality, we assume that the network $G$ is connected (if there is more than one connected component, we can consider each of them separately.) Moreover, we assume the topology of the sensor network is known (*e.g.*, the network is determined by design). In this paper, we attempt to study the spatial correlated data among the network (for temporal correlated, we can first build a time series model, and then consider the residual). Thus, we assume that sensor network $G$ monitors the sequences of independent observations across time in parallel. In the beginning, these observations follow a certain distribution $\mathcal{M}_1$. At an unknown time $\tau$, the populations change to another distribution $\mathcal{M}_2$. Specifically, there are $p$ streaming sequences $\boldsymbol{x}^t = (x_1^t, \cdots, x_p^t)^\top$ observed from $p$ sensors, where $x_i^t$ denotes the observation on sensor $i$ at time $t$. If there is no change occurring, *i.e.*, under the null hypothesis $\mathrm{H}_0$, the distribution

$\mathcal{M}_1$ has the following form:

$$\mathrm{H}_0: \quad \boldsymbol{x}^t = \boldsymbol{c}^t + \boldsymbol{\epsilon}^t \quad \overset{\text{i.i.d.}}{\sim} \quad \mathcal{N}(\boldsymbol{c}^t, \boldsymbol{\Sigma}), \quad \text{for } t = 1, 2, \cdots, \tag{5.2.1}$$

where $\boldsymbol{\epsilon}^t$ is a Gaussian noise with covariance matrix $\boldsymbol{\Sigma}$ and $\boldsymbol{c}^t$ is the mean of the observation. Here $\boldsymbol{c}^t$ is not necessary a constant; for example, in-control observations in a power network may follow certain wave patterns [163]. When there is a change on nodes $\mathcal{S}$ at an unknown time $\tau$, *i.e.*, under the alternative hypothesis $\mathrm{H}_A$, the distribution $\mathcal{M}_2$ follows:

$$\begin{aligned} \mathrm{H}_A: \boldsymbol{x}^t &= \boldsymbol{c}^t + \boldsymbol{\epsilon}^t \quad \overset{\text{i.i.d.}}{\sim} \quad \mathcal{N}(\boldsymbol{c}^t, \boldsymbol{\Sigma}), \quad \text{for } t = 1, 2, \cdots, \tau, \\ \boldsymbol{x}^t &= \boldsymbol{c}^t + b\boldsymbol{e}^{\mathcal{S}} + \boldsymbol{\epsilon}^t \quad \overset{\text{i.i.d.}}{\sim} \quad \mathcal{N}(\boldsymbol{c}^t + b\boldsymbol{e}^{\mathcal{S}}, \boldsymbol{\Sigma}), \quad \text{for } t = \tau+1, \tau+2, \cdots, \end{aligned} \tag{5.2.2}$$

where $b$ denotes for the mean shift magnitude, $\boldsymbol{e}^{\mathcal{S}} = (e_1^{\mathcal{S}}, \cdots, e_p^{\mathcal{S}})^\top$ denotes a 0-1 vector with $e_i^{\mathcal{S}} = 1$ if and only if $i \in \mathcal{S}$, and $\mathcal{S} \subset \{1, \cdots, p\}$ is an index set. Our goal is to design a decentralized monitor mechanism that utilizes the spatial correlation information and detects the unknown change time as quickly as possible, subject to constraints on the false alarm rate [164]. As a starting point for designing a decentralized detection method with correlated data, we assume that the post-change mean and the covariance matrix $\boldsymbol{\Sigma}$ are known.

**One-Shot.** In the literature, many works [148, 149, 150, 158] investigate the cases where the observations at each time are independent across nodes, that is $\boldsymbol{\Sigma}$ in (5.2.1) and (5.2.3) is a diagonal matrix. They then proposed to perform a detection on each single sensor, and once a sensor raises a local alarm, it transmits a one-bit signal to the fusion center. In practice, for the cases where $\boldsymbol{\Sigma}$ is not diagonal, we can still use the one-shot method by considering the network observations as the node-wise independent observations. Thus, we implement the one-shot method as our baseline by ignoring the spatial correlation among data. Moreover, since the in-control mean $\boldsymbol{c}^t$ might change, instead of directly monitoring the observations $\boldsymbol{x}^t$, we focus on the residuals as suggested by [165]. Specifically, we decompose the original alternative hypothesis into $p$ local alternative hypotheses $H_j, j = 1, ..., p$ for the residual $\boldsymbol{\epsilon}^t := \boldsymbol{x}^t - \boldsymbol{c}^t$ as follows:

$$
\begin{aligned}
\epsilon_j^t &\overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma_{jj}), \quad \text{for } t = 1, 2, \cdots, \tau, \\
\epsilon_j^t &\overset{\text{i.i.d.}}{\sim} \mathcal{N}(b e_j^{\mathcal{S}}, \Sigma_{jj}), \quad \text{for } t = \tau + 1, \tau + 2, \cdots,
\end{aligned} \tag{5.2.3}
$$

where $\epsilon_j^t$ is the $j$-th component of $\boldsymbol{\epsilon}^t$, $e_j^{\mathcal{S}}$ denotes the $j$-th element of $\boldsymbol{e}^{\mathcal{S}}$, and $\Sigma_{jj}$ denotes the $j$-th diagonal element of $\boldsymbol{\Sigma}$. Since in practice we do not know the set $\mathcal{S}$ in advance and we perform a local detection, we assume that a mean shift could happen on all nodes, *i.e.*, $e_j^{\mathcal{S}} = 1$ for all $j \in \{1, 2, \cdots, p\}$. We then derive the one-shot method with CUSUM [156]: Let $f_{\infty,i}(\cdot)$, $f_{0,i}(\cdot)$ denote the pre- and post-change probability density function (pdf) for marginal distributions of the observations on node $v_i$. Then the CUSUM statistics is defined by maximizing the log-likelihood ratio statistic over all possible change-point locations

$$
C_i^t = \max_{1 \le l \le t} \sum_{k=l}^{t} \log \frac{f_{0,i}(x_i^k)}{f_{\infty,i}(x_i^k)},
$$

which has the recursive implementation

$$
C_i^t = (C_i^{t-1})_+ + \log \frac{f_{0,i}(x_i^t)}{f_{\infty,i}(x_i^t)}.
$$

Therefore, under (5.2.3), we obtain:

$$
C_i^t = \max\left(C_i^{t-1} + \underbrace{\frac{b}{\Sigma_{ii}}\left(\epsilon_i^t - \frac{b}{2}\right)}_{\text{Log-likelihood Ratio of Marginal Distribution}}, 0\right).
$$

We further define the stopping time on node $v_i$ as:

$$
T_i(s) := \inf\left\{t : C_i^t \ge s\right\},
$$

where $s$ is the threshold for arising an alarm. Since in the one-shot scheme, a sensor will send a one-bit decision to the fusion center after it raises an alarm, we then have the stopping time for the whole network:

$$T(s) := \min_i \{T_i(s)\} = \inf \left\{ t : \max_i C_i^t \geq s \right\}. \tag{5.2.4}$$

The next Proposition characterizes the average running length and expected detection delay for the one-shot method with a single node change.

**Proposition 5.2.1 (Asymptotic Theory for One-Shot Method).** Suppose that without any change, $\boldsymbol{x}^t$ follows (5.2.1) and with change, $\boldsymbol{x}^t$ follows (5.2.3) for $\mathcal{S} = \{i\}$ for some i. Then as threshold $s \to \infty$, we have

1. Under null hypothesis $H_0$, the ARL of $T$ satisfies:

$$\mathbb{E}_\infty[T] \geq \frac{\exp(s)}{p} =: \gamma. \tag{5.2.5}$$

   Here $\mathbb{E}_\infty$ denotes the expectation without change.

2. Under alternative hypothesis $H_A$, the EDD of $T$ satisfies:

$$\mathbb{E}_i[T] \sim \frac{2\Sigma_{ii}}{b^2} \log(p\gamma). \tag{5.2.6}$$

   Here $\mathbb{E}_i[T] := \sup_{\tau \geq 1} (\text{ess sup}) \mathbb{E}_i^\tau \left[ (T - \tau + 1)^+ | \hat{\boldsymbol{\epsilon}}^1, \cdots, \hat{\boldsymbol{\epsilon}}^{\tau-1} \right]$ is the EDD of stopping time $T$, and $\mathbb{E}_i^\tau$ denotes the expectation with a change occurring on node $v_i$ at time $\tau$.

The proof of Proposition 5.2.1 is provided in Appendix E.1. Proposition 5.2.1 shows that the EDD for the one-shot method depends on the variance of the marginal distribution. This is quite reasonable since each node only uses its own data. However, this exactly is the drawback of the one-shot method, *i.e.*, lacking of *utilizing the neighbors' information*, which weakens the detection power. We will discuss this later in Section 5.3 in detail. We also remark that Proposition 5.2.1 can be further extended to multiple changed nodes as well as a general distribution change with some modification.

## 5.3 Conditional AutoRegressive Detection

In this section, we first review the Conditional AutoRegressive (CAR, [159]) model, then present our CAR detection (CARD) method, and establish the corresponding ARL and EDD for CARD. In the end, we show that CARD outperforms one-shot method by comparing two EDDs, and discuss the network architecture and correlation effects on the improvement.

**Conditional Auto-Regressive Model.** CAR models are often used to describe the spatial variation of quantities of interest in the form of summaries or aggregates over subregions. It is widely used to analyze data in diverse areas, such as demography, economy, epidemiology, and geography [166, 167]. The general goal of CAR models is to unveil and quantify spatial relations present among the data, in particular, to quantify how quantities of interest vary with explanatory variables and to detect clusters of "hot spots". Mathematically, a $0$-mean CAR model with Gaussian case is as follows:

$$\epsilon_i|\epsilon_j,\ j \neq i \sim \mathcal{N}\left(\phi \sum_{j\neq i} \frac{W_{ij}}{W_{i+}}\epsilon_j, \frac{\sigma^2}{W_{i+}}\right), \tag{5.3.1}$$

where $\boldsymbol{\epsilon} = (\epsilon_1, \cdots, \epsilon_p)^\top$ denotes the observation vector for all subregions, $\sigma^2$ denotes the shared varaiance for the network, $\boldsymbol{W} = (W_{ij}) \in \mathbb{R}^{p\times p}$ denotes the adjacency matrix[1] of graph $G$, $W_{i+} = \sum_{j=1}^p W_{ij}$ denotes the degree of node $v_i$, and $\phi$ is the correlation parameter controlling the spatial dependency: a large magnitude of $\phi$ means strong spatial dependency and a small one means weak dependency. Such a CAR model allows us to characterize the joint distribution of the network data via local conditional specification, as

[1]We can generalize the framework by extending adjacency matrix to proximity matrix, which reflects the "distance" between two subregions. See [168].

model (5.3.1) is equivalent to the following joint form by Brook's Lemma [160]:

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2(\boldsymbol{D_W} - \phi \boldsymbol{W})^{-1}\right), \tag{5.3.2}$$

where $\boldsymbol{D_W} = \mathrm{diag}(W_{1+}, \cdots, W_{p+})$. In order to make the model proper, *i.e.*, $\boldsymbol{D_W} - \phi\boldsymbol{W}$ is positive semi-definite, $\phi$ should satisfy $\frac{1}{\lambda_1} < \phi < \frac{1}{\lambda_p}$. Here $\lambda_1(\lambda_1 < 0)$ and $\lambda_p(\lambda_p > 0)$ are the smallest and largest eigenvalues of $\boldsymbol{D_W}^{-1/2}\boldsymbol{W}\boldsymbol{D_W}^{-1/2}$ respectively.

Note that in CAR model (5.3.1), all nodes share the same variance parameter $\sigma^2$, which is not very practical in many applications. For example, in a power network, different sensors may have different scales [163]. Therefore, instead of using the same $\sigma^2$, we adopt a variance structure from [169] as follows:

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}^{1/2}(\boldsymbol{D_W} - \phi \boldsymbol{W})^{-1}\boldsymbol{\Lambda}^{1/2}\right), \tag{5.3.3}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\sigma_1^2, \cdots, \sigma_p^2)$ denotes the variance scale matrix and $\sigma_i^2$ denotes the $i$-th node variance "scale". All these parameters $\phi$ and $\sigma_i'$s are estimable and the corresponding distributed algorithm is provided later in Section 5.4. Similar to (5.3.2), under model (5.3.3), for a specific node, given its neighbors' information, the conditional distribution is still a normal distribution:

$$\epsilon_i^t | \epsilon_j^t, j \neq i \sim \mathcal{N}\left(\frac{\phi}{W_{i+}} \sum_{j \in N(i)} \frac{\sigma_i}{\sigma_j}\epsilon_j^t, \frac{\sigma_i^2}{W_{i+}}\right). \tag{5.3.4}$$

Note that under model (5.3.4), the conditional mean of $\epsilon_i^t$ is the weighted sum of the observations for the neighbor subregions, which can be obtained by local communications. The following Lemma shows the likelihood ratio between the distribution under $H_A$ in (5.2.3) with $\mathcal{S} = \{i\}$ for some $i$, and the distribution under $H_0$ in (5.2.1).

**Lemma 5.3.1.** Suppose the observations follow the model in (5.3.4). At time $t$, the likelihood ratio between density $P_i$ under $H_A$ in (5.2.3) with $\mathcal{S} = \{i\}$ for some $i$, and density $P_\infty$ under $H_0$ in (5.2.1) is:

$$\frac{P_i(\boldsymbol{\epsilon}^t)}{P_\infty(\boldsymbol{\epsilon}^t)} = \exp\left(\frac{bW_{i+}}{\sigma_i^2}\Big(\epsilon_i^t - \frac{\phi}{W_{i+}}\sum_{j\in N(i)}\frac{\sigma_i}{\sigma_j}\epsilon_j^t - \frac{b}{2}\Big)\right). \tag{5.3.5}$$

The proof of Lemma 5.3.1 is provided in Appendix E.2. Lemma 5.3.1 implies that we can detection the change point by local monitoring. This allows us to implement a decentralized detection method, called Conditional AutoRegressive detection (CARD) method. Specifically, we compute the CUSUM statistics on node $v_i$ as follows:

$$\tilde{C}_i^t = \max\left(0, \tilde{C}_i^{t-1} + \underbrace{\frac{bW_{i+}}{\sigma_i^2}\Big(\epsilon_i^t - \frac{\phi}{W_{i+}}\sum_{j\in N(i)}\frac{\sigma_i}{\sigma_j}\epsilon_j^t - \frac{b}{2}\Big)}_{\text{Log-likelihood Ratio of Conditional Distribution}}\right). \tag{5.3.6}$$

Note that the CUSUM in (5.3.6) only uses the neighbors' information, which not only leverages the spatial correlation but can also be implemented in a decentralized system. Moreover, we adopt the same stopping rule as that of the one-shot method. Then the stopping time criterion for CARD is defined as:

$$\tilde{T}(s) = \inf_{i=1,\ldots p} \tilde{T}_i(s) \quad \text{where} \quad \tilde{T}_i(s) = \inf\left\{t : \tilde{C}_i^t \geq s\right\}. \tag{5.3.7}$$

The following theorem characterizes the ARL and the EDD for CARD.

**Theorem 5.3.2** (Asymptotic Theory for CARD)**.** Suppose that without change, the observation $\boldsymbol{x}^t$ follows (5.2.1), and with change, $\boldsymbol{x}^t$ follows (5.2.3) for $S = \{i\}$ for some $i$ and the noise follows (5.3.4). Then as threshold $s \to \infty$, we have

1. Under $H_0$ the average running length of $\tilde{T}$ satisfies:

$$\mathbb{E}_\infty[\tilde{T}] \geq \frac{\exp(s)}{p} =: \gamma. \tag{5.3.8}$$

2. Under alternative hypothesis $H_A$, the EDD of $\tilde{T}$ satisfies:

$$\mathbb{E}_i[\tilde{T}] \sim \frac{2s\sigma_i^2}{b^2 W_{i+}} = \frac{2\sigma_i^2}{b^2 W_{i+}} \log(p\gamma). \tag{5.3.9}$$

The proof of Theorem 5.3.2 is provided in Appendix E.3. Theorems 5.3.2 implies that as the threshold $s \to \infty$, the CARD has the same lower bound for ARL as the one-shot method has and the EDD for both methods are linear in $s$. We then compare their coefficients in their EDDs. The next proposition shows that if (5.3.4) holds, then CARD is more powerful than the one-shot method.

**Proposition 5.3.3.** With the same assumptions in Theorem 5.3.2, the following equality holds:

$$\frac{\mathbb{E}_i(T)}{\mathbb{E}_i(\tilde{T})} = \frac{\Sigma_{ii}}{\sigma_i^2/W_{i+}} = 1 + \underbrace{\sum_{l=1}^{\infty} \phi^l \sum_{(v_i, v_{j_1}, \cdots, v_{j_l}, v_i) \in Path_l^i} \frac{1}{W_{i+}} \prod_{k=1}^{l} \frac{1}{W_{j_k +}}}_{P_i(\phi)}, \tag{5.3.10}$$

where $Path_l^i = \{(v_i, v_{j_1}, \cdots, v_{j_l}, v_i) : (v_{j_k}, v_{j_{k+1}}) \in E, k = 0, \cdots, l, v_{j_0} = v_{j_{l+1}} = v_i\}$ is the set of length $l+1$ paths from $v_i$ to $v_i$. Moreover, the following inequality holds:

$$P_i(\phi) \geq \sum_{l=1}^{\infty} \left( \sum_{j \in N(i)} \frac{\phi^2}{W_{i+} W_{j+}} \right)^l \geq \frac{1}{1 - \frac{\phi^2}{W_{i^*+}}}. \tag{5.3.11}$$

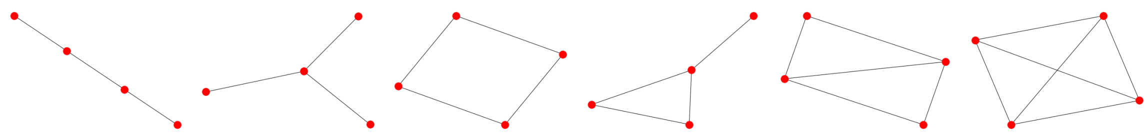
The proof of Proposition 5.3.3 is provided in Appendix E.4. Proposition 5.3.3 implies that the coefficient of EDD for CARD is smaller than that for the one-shot method, which shows the CARD is more powerful than the one-shot method. Moreover, Proposition 5.3.3 shows two key ingredients of the improvement: (1) correlation parameter $\phi$: the larger $\phi$ is, the more improvement CARD achieves; (2) connectivity: if a node has all neighbors with few degrees, then this node can gain more improvement. Based on these results, we then divide the nodes into three categories:

**1 Hub**: A node is called a hub if it has many neighbors and each of its neighbor is a spoke.

**2 Spoke**: A node is called spoke if it has a small degree.

**3 General**: A node is neither a hub nor a spoke.

In general, among these three-type nodes, the hub achieves most significant improvement, spoke has the least improvement, and the general node is in the middle.



(a) Line (b) Star (c) Circle (d) Paw (e) Diamond (f) $K_4$

Figure 5.2: An illustration of six network architectures for four nodes.

In addition, we study different four-node networks to illustrate the improvement. There are six architectures for a four-node network shown in Figure 5.2. We consider the following three criteria: (1) the average improvement over all nodes; (2) the robust improvement (the worst improvement among the nodes); (3) the targeted improvement (the best improvement among the nodes). The results are shown in Table 5.1. As can be seen, among these network architectures, (1) with the average criterion, the line network is the best; (2) with the robust criterion, the circle network is the best; (3) with the targeted criterion, a star network with this node as a center is the best. The calculation is provided in Appendix E.5.

| Architecture | Average Improvement | Robust Improvement | Targeted Improvement |
|---|---|---|---|
| Line | $\frac{5\phi^2-2\phi^4}{2(1-\phi^2)(4-\phi^2)}$ | $\frac{2\phi^2-\phi^4}{(1-\phi^2)(4-\phi^2)}$ | $\frac{3\phi^2-\phi^4}{(1-\phi^2)(4-\phi^2)}$ |
| Star | $\frac{\phi^2}{2(1-\phi^2)}$ | $\frac{\phi^2}{3(1-\phi^2)}$ | $\frac{\phi^2}{1-\phi^2}$ |
| Circle | $\frac{\phi^2}{2(1-\phi^2)}$ | $\frac{\phi^2}{2(1-\phi^2)}$ | $\frac{\phi^2}{2(1-\phi^2)}$ |
| Paw | $\frac{11\phi^2+3\phi^3-2\phi^4}{(2-\phi-\phi^2)[12+6\phi-2\phi^2]}$ | $\frac{2\phi^2-\phi^3}{(1-\phi)[6+3\phi-\phi^2]}$ | $\frac{4\phi^2-\phi^3}{(1-\phi)[6+3\phi-\phi^2]}$ |
| Diamond | $\frac{\phi^2(7+3\phi)}{2(9-7\phi^2+2\phi^3)}$ | $\frac{\phi^2}{(1-\phi)(3+2\phi)}$ | $\frac{4\phi^2+2\phi^3}{(1-\phi)(9+9\phi+2\phi^2)}$ |
| $K_4$ | $\frac{\phi^2}{(1-\phi)(3+\phi)}$ | $\frac{\phi^2}{(1-\phi)(3+\phi)}$ | $\frac{\phi^2}{(1-\phi)(3+\phi)}$ |

Table 5.1: The improvements that CARD achieves compared to the one-shot method under three criteria with different 4-node network architectures.

**Connection to Markov Chain.** The improvement $P_i(\phi)$ is also closely related to the Markov Chain. Let $(v_i, v_{j_1}, \cdots, v_{j_l}, v_i) \in Path_l^i$ denote a length $l+1$ non-simple path[2] from node $v_i$ and back to node $v_i$. For example, the shortest path from $v_i$ to $v_i$ is $(v_i, v_j, v_i)$, where $j \in N(i)$. Now let us consider a random walk over the network $G$ with transition matrix $\boldsymbol{D}_{\boldsymbol{W}}^{-1}\boldsymbol{W}$, *i.e.*, for each node $v_i$, it has an equal probability $\frac{1}{W_{i+}}$ to its neighbors. Then essentially, the right hand side of (5.3.10) is a probability generating function, which satisfies

$$P_i(\phi) = \frac{1}{1 - F_i(\phi)}, \quad \text{for } -1 < \phi < 1, \tag{5.3.12}$$

where $F_i(\phi) = \sum_{j=1}^{\infty} f_i^j \phi^j$ is the probability generating function of the hitting time

$$T_i := \inf\left\{t : s^t = i, s^k \neq i, 0 < k < j, s^0 = i\right\}$$

for the state $s$ starting from node $v_i$ and first ending at $v_i$, and

$$f_i^j := \mathbb{P}(s^j = i, s^k \neq i, 0 < k < j | s^0 = i)$$

denotes the probability for the state first returning to node $v_i$ at $j$-th step [171]. Since $F_1(1) = 1$ and $F_1(\phi)$ is monotone in $\phi$, the larger $\phi$ is, the more improvement CARD achieves.

## 5.4 Methodology

In this section, we discuss the methodological development for CARD, concerning the estimation of correlation parameter $\phi$ and variance scales $\sigma_i^2$, and then provide a full algorithmic statement.

A standard way to estimate parameters $\phi$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\sigma_1^2, \cdots, \sigma_p^2)$ is using maximum likelihood estimate (MLE) with some historical in-control data. For the standard CAR

[2] A path is simple if all of its vertices are distinct [170].

model with only one shared variance parameter $\sigma^2$ (*i.e.,* $\sigma_1^2 = \cdots = \sigma_p^2 = \sigma^2$), the MLEs are usually obtained through a two-step profile likelihood approach [172]. However, it requires the fusion center to collect all data over the whole network, which makes the data high-dimensional. Besides, the update of the correlation parameter $\phi$ needs to compute the determinant of $\boldsymbol{D_W} - \phi \boldsymbol{W}$, which makes the update extremely time consuming.

To address these issues, we propose a decentralized way to estimate these parameters efficiently. Instead of computing at the fusion center, we estimate the parameters, $\phi$ and $\boldsymbol{\Lambda}$ locally by leveraging the conditional specification of the CAR model, *i.e.,* each node $v_i$ would estimate $\sigma_i^2$ and $\phi$. To make it clear, we denote the estimate of $\phi$ on node $v_i$ as $\phi_i$. Recall that the conditional distribution for node $v_i$ is as follows:

$$\epsilon_i^t | \epsilon_j^t, j \neq i \sim \mathcal{N}\left(\frac{\phi^*}{W_{i+}} \sum_{j \in N(i)} \frac{\sigma_i^*}{\sigma_j^*} \epsilon_j^t, \frac{\sigma_i^{*2}}{W_{i+}}\right), \tag{5.4.1}$$

where $\phi^*$ and $\sigma_i^*$'s denote the ground truth correlation parameter and variance scales, respectively. Then we consider the following two-step procedure for each node to estimate the parameters via the least squares method in a distributed manner, and a full algorithm statement is present in Algorithm 6:

**Step 1. Estimate the variance "scales".** Node $v_i$ receives the $\epsilon_j'$s from its neighbors and then regresses $\epsilon_i$ on $\epsilon_j'$s. With the residual sum of squares, we can further obtain the estimate for $\sigma_i^{*2}$ given (5.4.1).

**Step 2. Estimate the correlation parameter.** After obtaining the variance scale estimates, we then plug them in objective (5.4.1) and obtain the MLE $\phi_i$ on node $v_i$. Each node then sends its local estimate to the central server. Then we can get a consensus estimate for $\hat{\phi}$ at the fusion center.

Here we first fit a model to predict the mean for the observation at each sensor, *e.g.*, it can be a constant or a time series model. Then we focus on the residual analysis [165]. We use the residual data to fit the parameters for the CAR model. After learning for the

**Algorithm 6** Learning Stage for In-control System

**Input:** Warm-up period: $T_w$; Variance-scale-estimate period: $T_\sigma$; Correlation-estimate period: $T_\phi$ In-control data: $\{x_i^t\}_{i=1}^p$, $t = 1, ..., T_w + T_\sigma + T_\phi$; Step sizes: $\eta_\phi, \eta_a$.
**Initialization:** $\phi_i^0 = 0$, $\sigma_i^0 = 1$, $a_i^j = 1$.
**Warm up** $t = 1, ..., T_w$**:** Each node $v_i$ builds a model $f_i(\cdot)$ to estimate $\hat{c}_i^t$ locally.
**for** $t \leftarrow T_w + 1$ *to* $T_w + T_\sigma$ *(Step 1)* **do**
  **parallel computing at nodes** $i = 1, \cdots, p$:
  Compute the residuals locally: $\hat{\epsilon}_i^t = x_i^t - \hat{c}_i^t$.
  Update the regression coefficients for neighbor $v_j$: $a_i^j = a_i^j + \eta_a \hat{\epsilon}_j^t \left(\hat{\epsilon}_i^t - \sum_{j\in N(i)} a_i^j \hat{\epsilon}_j^t\right)$.
  Update the variance estimates locally: $\hat{\sigma}_i^2 = \frac{t-T_w-1}{t-T_w}\hat{\sigma}_i^2 + \frac{1}{t-T_w}\left(\hat{\epsilon}_i^t - \sum_{j\in N(i)} a_i^j \hat{\epsilon}_j^t\right)^2$.
**for** $t \leftarrow T_w + T_\sigma + 1$ *to* $T_w + T_\sigma + T_\phi$ *(Step 2)* **do**
  **parallel computing at nodes** $i = 1, \cdots, p$:
  Compute the residuals locally: $\hat{\epsilon}_i^t = x_i^t - \hat{c}_i^t$.
  Update the correlation estimates locally:

$$\hat{\phi}_i = \hat{\phi}_i - \eta_\phi \frac{1}{W_{i+}} \sum_{j\in N(i)} \frac{\hat{\sigma}_i}{\hat{\sigma}_j}\epsilon_j^t \left(\epsilon_i - \frac{1}{W_{i+}} \sum_{j\in N(i)} \hat{\phi}_i \frac{\hat{\sigma}_i}{\hat{\sigma}_j}\epsilon_j^t\right).$$

Consensus on correlation $\phi$: $\hat{\phi} = \frac{1}{p}\sum_{i=1}^p \hat{\phi}_i$.
**Output:** correlation estimate $\hat{\phi}$, variance estimates $\hat{\sigma}_i^2$'s, and mean estimate models $f_i$'s.

**Algorithm 7** Monitoring Stage for Network System

**Input:** Correlation parameter estimate $\hat{\phi}$, Variance estimates $\hat{\sigma}_i^2$, Mean estimate models $f_i'$s, Streaming data $\{x_i^t\}_{i=1}^p$, $t = 1, 2, ...$, Control parameter $b$, and Threshold $s$.
**Initialization:** Cumulative sum $\tilde{C}_i^0 = 0$ for $i = 1, ..., p$, time $t = 0$.
**while** $\max_i \tilde{C}_i^t \leq s$ **do**
  **Time update:** $t = t + 1$.
  **parallel computing at nodes** $i = 1, \cdots, p$:
  Obtain the mean estimate from $f_i$: $\hat{c}_i^t$.
  Compute the residual locally: $\hat{\epsilon}_i^t = x_i^t - \hat{c}_i^t$.
  Updates the local CUSUM

$$\tilde{C}_i^t = \max\left(0, \tilde{C}_i^{t-1} + \frac{bW_{i+}}{\hat{\sigma}_i^2}\left(\hat{\epsilon}_i^t - \frac{\hat{\phi}}{W_{i+}} \sum_{j\in N(i)} \frac{\hat{\sigma}_i}{\hat{\sigma}_j}\hat{\epsilon}_j^t - \frac{b}{2}\right)\right).$$

**Raise alert:** nodes whose CUSUMs exceed threshold $s$ send alarm to the fusion center.

in-control system, we then use the learnt model to monitor the whole system and the full algorithm statement is provided in Algorithm 7. As can be seen, both learning and monitoring stages for the network system are decentralized except the consensus for the correlation parameter $\phi$. This shows that CARD can be implemented for a large scale network and monitor the system effectively and efficiently.

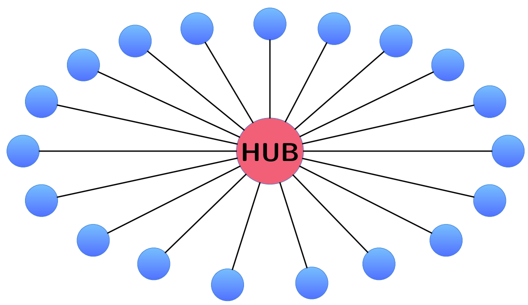
## 5.5 Numerical Simulation

In this section, we conduct numerical simulations to compare CARD and one-shot methods with two network architectures: Star network [173] and Erdős-Rényi network [174]. Here we assume that all nodes share the same variance parameter.

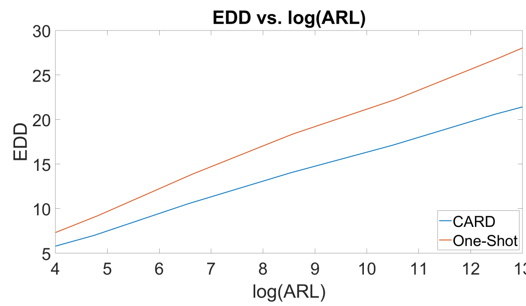
**Star Network.** A 20-node star network, shown in Figure 5.3a, contains one hub (red) and 19 spokes (blue). As for the data, we generate the synthetic data from a CAR model with $\phi = 0.5$, noise $\sigma^2 = 4$, and mean shift $b = 0.5$. We compare two different settings: (1) hub as the changed node; (2) spoke as the changed node. We repeat both settings for 600 times. Figures 5.3b and 5.3c present the mean of the simulation results and illustrate the relationship between the EDD and $\log$(ARL) for CARD and the one-shot method. As we can see, for hub as the changed node, the CARD outperforms the one-shot method significantly; while for spoke as the changed node, the improvement is less significant, which matches our theoretical results.

**Erdős-Rényi Network**: First we generate 30 random networks with 20 nodes and $\rho = 0.7$, where $\rho$ is the probability for each edge appearing in the network. By [174], such networks are almost surely connected. Then for each network, we run 200 simulations and in every simulation we randomly assign a changed node. As for the data, we adopt the same setting as that in the star network experiment, that is, we generate the synthetic data from a CAR model with $\phi = 0.5$, noise $\sigma^2 = 4$, and mean shift $b = 0.5$. Figure 5.3d presents the mean of the simulation results. As we can see, the slope of CARD is smaller and it becomes more significant when ARL increases.

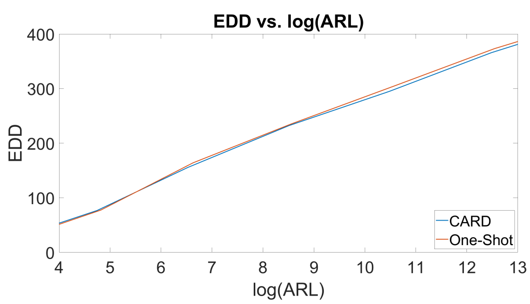
As shown in Figure 5.3, the improvement for the hub is the most significant as shown in Figure 5.3b; the improvement for the spoke is the least as shown in Figure 5.3c; for general cases, the improvement is between those for the spoke and for the hub as shown in Figure 5.3d, which matches our intuition in Section 5.3. Moreover, note that results in Proposition 5.2.1 and Theorem 5.3.2 are both asymptotic, which explains the phenomena that the difference become more significant when the ARL increases in Figures 5.3b-5.3d.



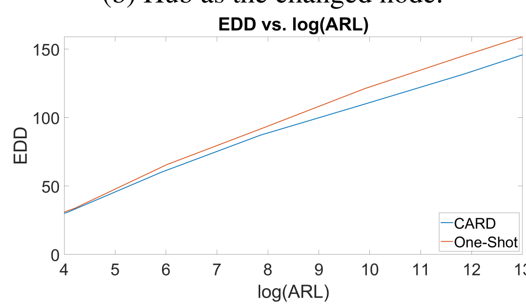

(a) Illustration of the star network.




(b) Hub as the changed node.




(c) Spoke as the changed node.



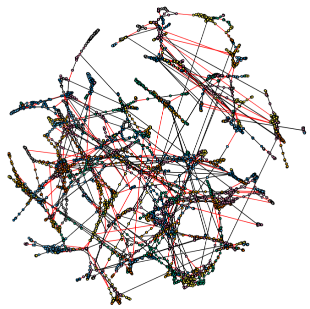

(d) Random change in Erdős Rényi with $\rho = 0.7$.

Figure 5.3: Network structures and simulation results under Star and Erdős-Rényi settings. (a) demonstrates a 20-node star network. In (b)-(d), y-axis is the EDD and x-axis is the ARL in log scale. The slope of curve reflects the power of detection method, the smaller the better.

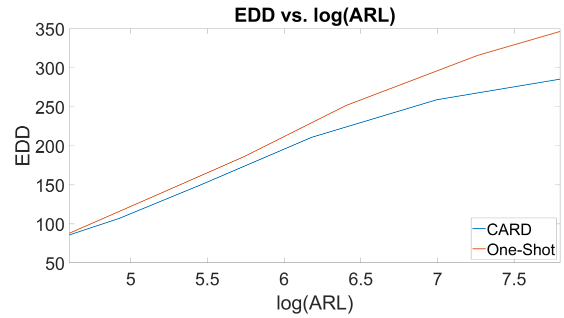
## 5.6 Two Applications

### 5.6.1 Western states power grid monitoring

In this section, we run a simulation with synthetic data generated on a real power network topology. Here we consider the Western States Power Grid of the USA, the states west of the rocky mountains. It consists of 4941 nodes and 6594 edges [175], in which the nodes are transformers, substations, and generators, and edges are high-voltage transmission lines shown in Figure 5.4a.



(a) Visualization for Western states power grid.




(b) Simulation results for power grid.

Figure 5.4: Simulation with synthetic data over Western States Power Grid of the USA.

In this application, we simulate a situation in which a power failure occurs over this large network. Assume that at each time, we observe the real power injection at an edge. When the power system is in a steady state, the observation is the true state plus Gaussian observation noise [176]. We then estimate the true state (*e.g.*, using techniques in [176]), subtract it from the observation value, and obtain the residual value on each node, which can be assumed to follow a CAR model with different variance scales. When the failure happens in a power system, there will be a mean shift over one node, since in practice, if we have a perfect time series model, when there is a power failure, usually only one residual will have the mean shift.

To compare the methods, we randomly choose one changed node over the network as what we did in the Erdős-Rényi experiment. We repeat the simulation for 600 times and present the results of EDD and ARL for the methods shown in Figure 5.4b. As can be seen, for large ARL, the slope of CARD is smaller than that of the one-shot method, *i.e.,* a clear improvement gained by CARD. Here for illustration, we consider the case of a single changed node. With such a simplified model for power networks, we aim to shed some light on the potential of CARD when applied to monitoring real power networks. We remark that CARD can be applied to the situations with multiple changed nodes, which we shall consider in application.

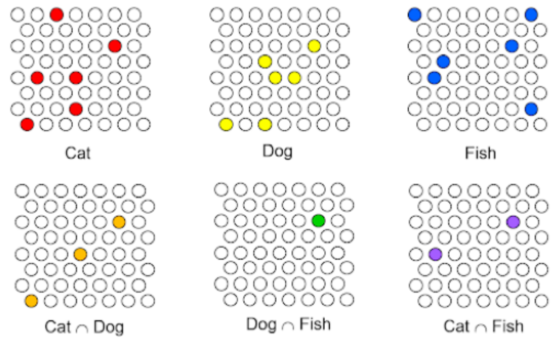


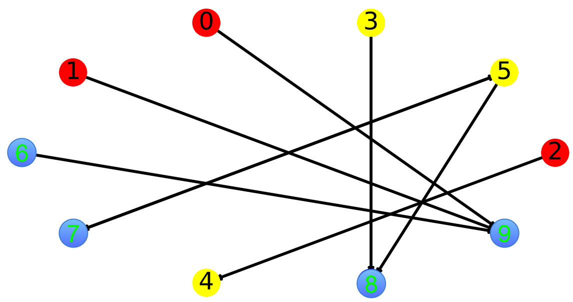

Figure 5.5: Illustrative examples of different states for biological neural networks with a cat, a dog and a fish as an external stimulus. The active nodes are marked by colors and similar signals shares more common nodes.

### 5.6.2 Stimulus detection in nervous systems

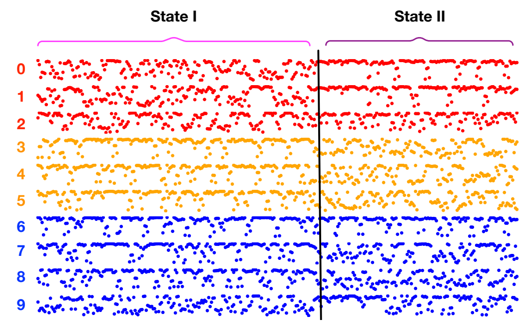
Change-point detection is an important problem in neuroscience: it is extremely useful for researchers to identify when an external stimulus is internalized as information in a biological neural network. In human brain, there are approximately 100 billion neurons, and the fundamental unit of information is the spike of a neuron's voltage. Thus, the biological neural networks are not easily readable, even with modern neuroimaging methods.

[177] found that cortex represents the information by sparse population coding: at any given point in time, only approximately 2% of neurons in human brain are active (high frequency in having a spike), and the activity of this subset of neurons collectively encodes representations of external stimuli, (*i.e.,* different information). Those external stimuli drive the network to different states shown in Figure 5.5. Therefore, we can find the stimulus change by monitoring the states of neurons.

For our purposes, our experiment assumes that we are observing a small patch of higher order cortex, and aim to identify the change of the network states. Our data is generated by the PyNN package [178] in conjunction with the NEURON simulator [179]. For illustration, we simulate a network of neurons with 10 neurons shown in Figure 5.6a. From the simulation, we can observe a continuous readout of each neuron's voltage, allowing us to utilize sub-threshold information in addition to traditional spiking information. A selected few neurons receive inputs from an external source, which represents the phenomenon of sparse population coding. The neurons that spike at higher rates form a distributed representation of a state.



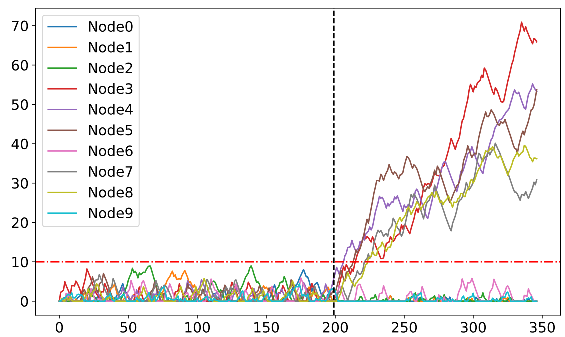

(a) Visualization for the biological neural network.



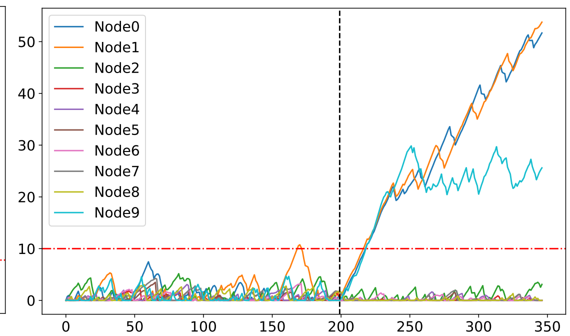

(b) Visualization for the test data.

Figure 5.6: An illustration for the sparse population coding. (a): The simulated biological neural network with excitatory neurons. (b): Simulated data: in each state, some neurons are stimulated by an external stimulus, and then their descendants become active. (The data for active nodes is more random in (b)) For example, in state I, neurons 0-2 are stimulated by an external stimulus and then neuron 9 also becomes active; in state II, neurons 3-5 are stimulated and then neurons 7 and 8 become active. The shift of these two states is marked by a black vertical line in (b).

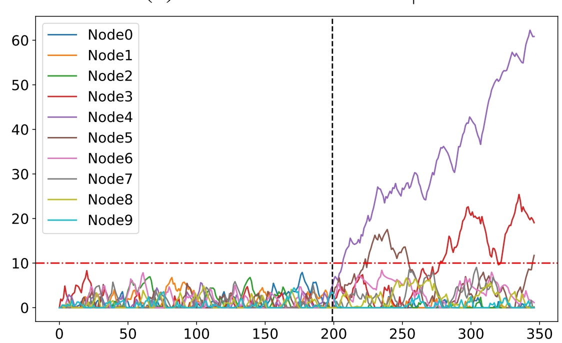
In our simulation, we randomly select two sets of neurons, representing two different states: the neurons in the first set and in the second set are stimulated in the states I and II, respectively. For the data, instead of using the original voltage data, we use the rolling window autocorrelation of the voltage data as suggested by [180]. We use some extra in-control data to learn the structure of the network under a specific population coding. After the structure and activity are well captured by the model, we conduct a test of the stimulus change-point detection algorithm by simulating a change from the first state to the second state. Figure 5.6b demonstrates the test data that we generated. We then apply two methods, the one-shot method and CARD, to find the changed nodes. The simulation results are presented in Figure 5.7. As can be seen, with threshold 10, both methods have the same average running length, 175, and expected detection delay, 6. Moreover, for the one-shot, it finds most changed nodes including those descendants; however it misses node 2, which is the most important one as it is directed stimulated by external signals. In contrast, CARD finds all directly stimulated nodes, *i.e.*, nodes 0-5. Note that CARD, by leveraging the spatial correlation, can identify node 2, which has a relatively low signal to noise ratio. This simulation study suggests that both methods are good for the multiple changed nodes and CARD can find the root cause nodes even with a relatively low signal to noise ratio.



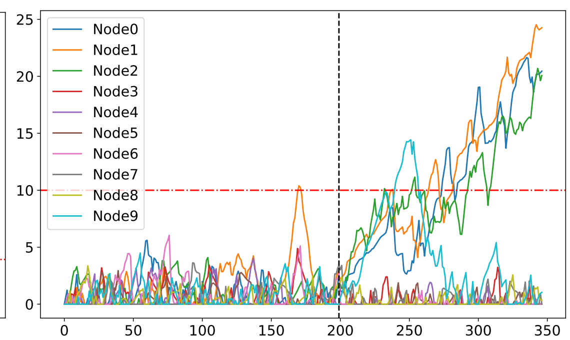

(a) One-Shot CUSUM$_+$




(b) One-Shot CUSUM$_-$




(c) CARD CUSUM$_+$




(d) CARD CUSUM$_-$

Figure 5.7: The monitor statistics with different methods. The vertical line denotes the iteration in which state changes and the horizontal line denotes the threshold. In practice, since we do not know the mean shift $b$ is positive or negative, we use both positive CUSUMs ((a) and (c)) and negative CUSUMs ((b) and (d)).

## 5.7 Conclusion

In this paper, to break the stringent independence assumption in the network change-point problem and leverage the spatial correlation, we propose a new model-based monitoring framework, Conditional AutoRegressive Detection (CARD), which models spatial correlations over the network via a Conditional AutoRegressive (CAR) model. We show that the conditional specification of the CAR model allows for a decentralized detection method which leverages spatial correlations by utilizing neighborhood information on each node. Theoretically, we prove that the expected detection delay for CARD is smaller than that for the one-shot method, which shows the improved detection power of the proposed method. We then demonstrate the improved detection performance of CARD over existing methods in a suite of numerical simulations and two applications: power grid monitoring, and sparse population coding of neuronal networks.

# Appendices

# APPENDIX A

# SUPPLEMENTARY MATERIALS IN CHAPTER 1

## A.1 Detailed Proofs in Section 1.3

### A.1.1 Proof of Proposition 1.3.3

*Proof.* We consider a compact singular value decomposition of $\Sigma_{XY}$ as follow:

$$\Sigma_{XY} = \sum_{i=1}^{r} \lambda_i \overline{u}_i \overline{v}_i^\top,$$

where $\lambda_1 > \lambda_2 \geq ... \geq \lambda_r > 0$ are nonzero singular values, and $(\overline{u}_i,\ \overline{v}_i)$'s are a pair of singular vectors associated with $\lambda_i$. Plugging (1.2.4) into (1.2.3), we have

$$\Sigma_{XY} v - (u^\top \Sigma_{XY} v) u = 0 \quad \text{and} \quad \Sigma_{XY}^\top u - (u^\top \Sigma_{XY} v) v = 0. \tag{A.1.1}$$

Since every vector $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^d$ can be expanded as

$$u = \sum_{i=1}^{r} c_i \overline{u}_i + \sum_{j=r+1}^{m} c_j \overline{u}_j \quad \text{and} \quad v = \sum_{i=1}^{r} l_i \overline{v}_i + \sum_{j=r+1}^{d} l_j \overline{v}_j, \tag{A.1.2}$$

where $\overline{u}_j$ for $j = r+1, ..., m$ and $\overline{v}_j$ for $j = r+1, ..., d$ are orthonormal basis vectors, and complementary to $\overline{u}_i$'s and $\overline{v}_i$'s for $i = 1, ..., r$ in $\mathbb{R}^m$ and $\mathbb{R}^d$ respectively, and $c_i$'s and $l_i$'s are the coefficients. Plugging (A.1.2) into the first equation of (A.1.1), we get

$$
\begin{aligned}
0 &= \sum_{i=1}^{r} \lambda_i \overline{u}_i \overline{v}_i^\top \cdot \sum_{i=1}^{d} c_i \overline{v}_i - \sum_{i=1}^{m} l_i \overline{u}_i \cdot \sum_{i=1}^{r} \lambda_i \overline{u}_i \overline{v}_i^\top \cdot \sum_{i=1}^{d} c_i \overline{v}_i \cdot \sum_{i=1}^{m} l_i \overline{u}_i \\
&= \sum_{i=1}^{r} c_i \lambda_i \overline{u}_i - \sum_{i=1}^{m} \big( \sum_{k=1}^{r} l_k \lambda_k c_k \big) \cdot l_i \overline{u}_i \\
&= \sum_{i=1}^{r} \left( c_i \lambda_i - \big( \sum_{k=1}^{r} l_k \lambda_k c_k \big) \cdot l_i \right) \overline{u}_i - \sum_{i=r+1}^{m} \big( \sum_{k=1}^{r} l_k \lambda_k c_k \big) \cdot l_i \overline{u}_i. \qquad \text{(A.1.3)}
\end{aligned}
$$

The second equality holds because $\overline{u}_i$ and $\overline{v}_j$ are the columns of the orthogonal matrices. Since $\overline{u}_i$'s are the basis vectors of $\mathbb{R}^m$, by (A.1.3), we know the coefficients of all $\overline{u}_i$'s should be $0$. Therefore we consider two scenarios:

1. If $\sum_{i=1}^{r} l_k \lambda_k c_k = 0$, then we have $c_i = 0$, $i = 1, 2, ..., r$. Similarly, plugging (A.1.2) into the second equation of (A.1.1), we have $l_i = 0$, $i = 1, 2, ..., r$. Thus, $u$ and $v$ are in the row and column null space of $\Sigma_{XY}$ respectively.

2. If $\sum_{i=1}^{r} l_k \lambda_k c_k \neq 0$, then we have $l_i = 0$, $i = r+1, ..., m$, which further leads to:

$$
c_i \lambda_i = \big( \sum_{k=1}^{r} l_k \lambda_k c_k \big) \cdot l_i \quad \text{and} \quad l_i \lambda_i = \big( \sum_{k=1}^{r} l_k \lambda_k c_k \big) \cdot c_i \quad \text{for } i = 1, 2, ..., r. \qquad \text{(A.1.4)}
$$

   Note that (A.1.4) holds if and only if there exists only one $i \in \{1, 2, ..., r\}$. $c_j = l_j = \pm\delta_{ij},\ j = 1, 2, ..., r$, where $\delta_{ij}$ is the Kronecker delta, i.e., $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$.

The verification of the above points satisfying (A.1.1) is straightforward, and therefore omitted. $\square$

### A.1.2 Proof of Proposition 1.3.4

*Proof.* For notation simplicity, we denote $\nabla^2_{u,v}L(u,v)$ as $\nabla^2_{u,v}L(u,v,\mu,\sigma)\Big|_{\mu=\sigma=\frac{1}{2}u^\top Av}$

$$\nabla^2_{u,v}L(u,v) = \begin{pmatrix} -u^\top \Sigma_{XY} v \cdot I_m & \Sigma_{XY} \\ \Sigma_{XY}^\top & -u^\top \Sigma_{XY} v \cdot I_d \end{pmatrix}.$$

**a.** If $u$ and $v$ are in the row and column null space of $\Sigma_{XY}$ respectively, then

$$\nabla^2_{u,v}L(u,v) = \begin{pmatrix} 0 & \Sigma_{XY} \\ \Sigma_{XY}^\top & 0 \end{pmatrix} \quad \text{and} \quad \lambda_{\max}(\nabla^2_{u,v}L(u,v)) = \lambda_1.$$

Therefore, it is an unstable stationary point because of the positive curvature.

**b.** If $(u,v)$ is a pair of singular vector of $\lambda_i$, then by simple linear algebra, we know that

$$\nabla^2_{u,v}L(u,v) \sim \begin{pmatrix} -u^\top \Sigma_{XY} v \cdot I_m & 0 \\ 0 & \frac{1}{u^\top \Sigma_{XY} v}\Sigma_{XY}^\top \Sigma_{XY} - u^\top \Sigma_{XY} v \cdot I_d \end{pmatrix}.$$

One can verify

$$\lambda_{\max}(\nabla^2_{u,v}L(u,v)) = \frac{\lambda_1^2 - \lambda_i^2}{\lambda_i} \geq \lambda_1 - \lambda_2.$$

Therefore, the Hessian matrix is negative semi-definite if and only if $u^\top \Sigma_{XY} v = \lambda_1$, i.e., $(u,v)$ is the optimum of (1.1.1). The Hessian has a positive eigenvalue.

Thus, only the optima of (1.2.2) are stable stationary points. All the others are unstable. □

## A.2 Detailed Proofs in Section 1.4

### A.2.1 Proof of Theorem 1.4.4

*Proof.* First, we calculate the infinitesimal conditional expectation. Since the optimization problem is symmetric about $u$ and $v$, we only prove the claim for $u$,

$$\begin{aligned}\frac{d}{dt}\mathbb{E}\left(U_\eta(t)-U_\eta(0)\right)\big|_{t=0} &= \eta^{-1}\mathbb{E}\left(U_\eta(\eta)-U_\eta(0)\big|U_\eta(0),V_\eta(0)\right)\\ &= \Sigma_{XY}V(0)-U(0)^\top\Sigma_{XY}V(0)U(0).\end{aligned}$$

Next, we show that if the initial is on the sphere, then with probability $1$, all iterations are on the sphere as $\eta\to 0^+$. Given $\|u_k\|_2=\|v_k\|_2=1$, we have

$$\begin{aligned}\|u_{k+1}\|_2^2 &= \left(u_k+\eta\cdot(X_kY_k^\top v_k-u_k^\top X_kY_k^\top v_ku_k)\right)^\top\cdot\left(u_k+\eta\cdot(X_kY_k^\top v_k-u_k^\top X_kY_k^\top v_ku_k)\right)\\ &= u_k^\top u_k+2\eta(u_k^\top X_kY_k^\top v_k-u_k^\top X_kY_k^\top v_ku_k^\top u_k)+\eta^2\|X_kY_k^\top v_k-u_k^\top X_kY_k^\top v_ku_k\|_2^2\\ &= 1+\eta^2\|X_kY_k^\top v_k-u_k^\top X_kY_k^\top v_ku_k\|_2^2.\end{aligned}$$

Therefore, we get

$$\mathbb{P}\Big(\lim_{\eta\to 0^+}\|u_{k+1}\|_2=1\Big|\|u_k\|_2=1\Big)=\mathbb{P}(|X_k^\top Y_k|<\infty)=1.$$

The last equality holds, since $\mathbb{E}|X_k^\top Y_k|$ is finite:

$$\mathbb{E}|X_k^\top Y_k|\le\sqrt{\mathbb{E}\|X_k\|_2^2\cdot\mathbb{E}\|Y_k\|_2^2}\le B^2d.$$

Finally, we bound the infinitesimal conditional variance.

$$
\begin{aligned}
&\frac{d}{dt}\mathbb{E}\left(U_\eta^{(j)}(t)-U_\eta^{(j)}(0)\right)^2\Big|_{t=0}\\
&\leq \eta^{-1}\cdot \mathrm{tr}\left(\mathbb{E}\left[\left(U_\eta(\eta)-U_\eta(0)\right)\left(U_\eta(\eta)-U_\eta(0)\right)^\top\right)\Big|U_\eta(0)=u_k,\ V_\eta(0)=v_k\right]\\
&=\eta^{-1}\cdot\mathbb{E}\left[\eta\left(X_kY_k^\top u_k-u_k^\top X_kY_k^\top v_ku_k\right)^\top\cdot\eta\left(X_kY_k^\top u_k-u_k^\top X_kY_k^\top v_ku_k\right)\right]\\
&=\eta\cdot\mathbb{E}\left(u_k^\top Y_kX_k^\top X_kY_k^\top u_k-2u_k^\top Y_kX_k^\top u_ku_k^\top X_kY_k^\top v_k+u_k^\top u_k(u_k^\top X_kY_k^\top v_k)^2\right)\\
&\leq \eta\cdot\left(\sqrt{\mathbb{E}\|X_k\|_2^4\mathbb{E}\|Y_k\|_2^4}+2\sqrt{\mathbb{E}(u_k^\top Y_kX_k^\top u_k)^2\mathbb{E}(u_k^\top Y_kX_k^\top v_k)^2}+\mathbb{E}(u_k^\top X_kY_k^\top v_k)^2\right)\\
&\leq \eta\cdot\left(\sqrt{\mathbb{E}\|X_k\|_2^4\mathbb{E}\|Y_k\|_2^4}+3\mathbb{E}(|Y_k^\top||X_k|)^2\right)\\
&=O(\eta).
\end{aligned}
$$

Last equality holds by the Assumption 1.4.1.

Therefore, by Section 4 of Chapter 7 in [17], we know that, as $\eta\to 0^+$, $U_\eta(t)$ and $V_\eta(t)$ weakly converge to the solution of (1.4.1) with the same initial. By definition of $U_\eta(t)$ and $V_\eta(t)$, we complete the proof. □

#### A.2.2 Proof of Theorem 1.4.5

*Proof.* Since $P$ is an orthonormal matrix, $\|H_j\|_2=\|W_j\|_2=1$ for all $j=1,...,d$. Thus, we have

$$
\begin{aligned}
\frac{d}{dt}H^{(i)}&=\lambda_iH^{(i)}-\sum_{j=1}^{2d}\lambda_j(H^{(j)})^2H^{(i)}\\
&=\lambda_i\sum_{j=1}^{2d}(H^{(j)})^2H^{(i)}-\sum_{j=1}^{2d}\lambda_j(H^{(j)})^2H^{(i)}\\
&=H^{(i)}\sum_{j=1}^{2d}\left(\lambda_i-\lambda_j\right)(H^{(j)})^2.
\end{aligned}
$$

We then verify (1.4.7) satisfies (1.4.6). By [181], we know that since $H_j(t)$ is continuously differentiable in $t$, the solution to the ODE is unique. For notational simplicity, we denote

$$S^{(j)}(t)=H^{(j)}(0)\exp(\lambda_jt).$$

Then we have

$$H^{(i)}(t) = \frac{S^{(i)}(t)}{\sqrt{\sum_{j=1}^{2d}\left(S^{(j)}(t)\right)^2}}.$$

Now we only need to verify

$$\begin{aligned}
\frac{d}{dt}H^{(i)}(t) &= \frac{\left(\lambda_i S^{(i)}(t)\right)\sqrt{\sum_{j=1}^{2d}\left(S^{(j)}(t)\right)^2} - \frac{\left(2\sum_{j=1}^{2d}\lambda_j\left(S^{(j)}(t)\right)^2\right)S^{(i)}(t)}{2\sqrt{\sum_{j=1}^{2d}\left(S^{(j)}(t)\right)^2}}}{\sum_{j=1}^{2d}\left(S^{(j)}(t)\right)^2} \\
&= \lambda_i \frac{S^{(i)}(t)}{\sqrt{\sum_{j=1}^{2d}\left(S^{(j)}(t)\right)^2}} - \sum_{j=1}^{2d}\lambda_j \frac{\left(S^{(j)}(t)\right)^2}{\sum_{j=1}^{2d}\left(S^{(j)}(t)\right)^2}\frac{S^{(i)}(t)}{\sqrt{\sum_{j=1}^{2d}\left(S^{(j)}(t)\right)^2}} \\
&= \lambda_i H^{(i)}(t) - \sum_{j=1}^{2d}\lambda_j\left(H^{(j)}(t)\right)^2 H^{(i)}(t),
\end{aligned}$$

which completes the proof. □

## A.3 Detailed Proofs in Section 1.5

### A.3.1 Proof of Theorem 1.5.1

*Proof.* We prove this by contradiction. Assume the conclusion does not hold, that is there exists a constant $C > 0$, such that for any $\eta' > 0$ we have

$$\sup_{\eta \le \eta'} \mathbb{P}(\sup_t |Z_\eta^{(i)}(t)| \le C) = 1.$$

That implies there exists a sequence $\{\eta_n\}_{n=1}^\infty$ converging to $0$ such that

$$\lim_{n\to\infty} \mathbb{P}(\sup_t |Z_{\eta_n}^{(i)}(t)| \le C) = 1. \tag{A.3.1}$$

Thus, condition (i) in Theorem 2.4 [182] holds. We next check the second condition. When $\sup_t |Z_{\eta_n}^{(i)}(t)| \le C$ holds, Assumption 1.4.1 yields that $z_{\eta_n,k+1}^{(i)} - z_{\eta_n,k}^{(i)} = C'\eta_n$, where $C'$ is some constant. Thus, for any $t, \epsilon > 0$, we have

$$|Z_{\eta_n}^{(i)}(t) - Z_{\eta_n}^{(i)}(t+\epsilon)| = \frac{\epsilon}{\eta} C' \eta = C'\epsilon.$$

Thus, condition (ii) in Theorem 2.4 [182] holds. Then we have $\{Z_{\eta_n}^{(i)}(\cdot)\}_n$ is tight and thus converges weakly.

We then calculate the infinitesimal conditional expectation and variance for $Z_{\eta_n}^{(i)},\ i \neq j$.

$$\begin{aligned}
\frac{d}{dt}\mathbb{E}Z_{\eta_n}^{(i)}(t)\big|_{t=0} &= \eta_n^{-1}\mathbb{E}\left[Z_{\eta_n}^{(i)}(\eta_n) - Z_{\eta_n}^{(i)}(0)\big|H_{\eta_n}(0) = h\right] \\
&= \eta_n^{-1}\mathbb{E}\left[\eta_n^{-1/2}\left(H_{\eta_n}^{(i)}(\eta_n) - H_{\eta_n}^{(i)}(0)\right)\big|H_{\eta_n}(0) = h\right] \\
&= \eta_n^{-1/2} h^{(i)} \sum_{l=1}^{2d} (\lambda_i - \lambda_l)\,(h^{(l)})^2 = Z_{\eta_n}^{(i)}\,(\lambda_i - \lambda_j) + o(1), \qquad \text{(A.3.2)}
\end{aligned}$$

where the last equality comes from the assumption that the algorithm starts near $j^{th}$ column of $P,\ j \neq 1$, i.e., $h \approx e_j$. To compute variance, we first compute $\hat{\Lambda}$,

$$\hat{\Lambda} = P^\top Q P = \frac{1}{2}\begin{pmatrix} \overline{Y}\,\overline{X}^\top + \overline{X}\,\overline{Y}^\top & \overline{Y}\,\overline{X}^\top - \overline{X}\,\overline{Y}^\top \\ -\overline{Y}\,\overline{X}^\top + \overline{X}\,\overline{Y}^\top & -\overline{Y}\,\overline{X}^\top - \overline{X}\,\overline{Y}^\top \end{pmatrix},$$

where $Q$ is defined in (1.4.2). Then we analyze $e_i^\top \hat{\Lambda} e_j$ by cases:

$$e_i^\top \hat{\Lambda} e_j = \begin{cases} \frac{1}{2}\left(\overline{X}^{(i)}\overline{Y}^{(j)} + \overline{X}^{(j)}\overline{Y}^{(i)}\right) & \text{if } \max(i,j) \leq d, \\ \frac{1}{2}\left(-\overline{X}^{(j)}\overline{Y}^{(i-d)} + \overline{X}^{(i-d)}\overline{Y}^{(j)}\right) & \text{if } j \leq d < i, \\ \frac{1}{2}\left(\overline{X}^{(j-d)}\overline{Y}^{(i)} - \overline{X}^{(i)}\overline{Y}^{(j-d)}\right) & \text{if } i \leq d < j, \\ \frac{1}{2}\left(-\overline{X}^{(i-d)}\overline{Y}^{(j-d)} - \overline{X}^{(j-d)}\overline{Y}^{(i-d)}\right) & \text{if } \min(i,j) > d, \end{cases}$$

which further implies

$$
\begin{aligned}
&\frac{d}{dt}\mathbb{E}(Z^{(i)}_{\eta_n}(t) - Z^{(i)}_{\eta_n}(0))^2\Big|_{t=0} \\
=&\eta_n^{-1}\mathbb{E}\big[\big(Z^{(i)}_{\eta_n}(\eta) - Z^{(i)}_{\eta_n}(0)\big)^2\big|H_{\eta_n}(0) = h\big] \\
=&\eta_n^{-2}\mathbb{E}[\eta_n^2(\hat{\Lambda}h - h^\top\hat{\Lambda}hh)(\hat{\Lambda}h - h^\top\hat{\Lambda}hh)^\top]_{i,i} \\
=&\mathbb{E}(e_i^\top\hat{\Lambda}e_je_j^\top\hat{\Lambda}^\top e_i) + o(1) \\
=&\frac{1}{4}\big(\gamma_i\omega_j + \gamma_j\omega_i + 2\,\mathrm{sign}(i - d - 1/2)\cdot\mathrm{sign}(j - 1/2 - d)\cdot\alpha_{ij}\big). \qquad \text{(A.3.3)}
\end{aligned}
$$

By (A.3.2) and (A.3.3), we get the limit stochastic differential equation,

$$dZ^{(i)}(t) = -(\lambda_j - \lambda_i)Z^{(i)}(t)dt + \beta_{ij}dB(t).$$

Therefore, $\{Z^{(i)}_{\eta_n}(\cdot)\}$ converges weakly to a solution of The process defined by the equation above is an unstable O-U process with mean $0$ and exploding variance. Thus, for any $\tau$, there exist a time $t'$, such that

$$\mathbb{P}(|Z^{(i)}(t')| \geq C) \geq 2\tau.$$

Since $\{Z^{(i)}_{\eta_n}\}_n$ converges weakly to $Z^i$, thus $\{Z^{(i)}_{\eta_n}(t')\}_n$ converges in distribution to $Z^i(t')$. This implies that there exists an $N > 0$, such that for any $n > N$

$$|\mathbb{P}(|Z^i(T)| \geq C) - \mathbb{P}(|Z^{(i)}_{\eta_n}(T)| \geq C)| \leq \tau.$$

Then we find a $t'$ such that

$$\mathbb{P}(|Z^{(i)}_{\eta_n}(t')| \geq C) \geq \tau, \forall n > N,$$

or equivalently

$$\mathbb{P}(|Z^{(i)}_{\eta_n}(t')| \le C) < 1-\tau, \forall n > N.$$

Since $\left\{\omega \middle| \sup_t |Z^{(i)}_{\eta_n}(t)(\omega)| \le C\right\} \subset \left\{\omega \middle| |Z^{(i)}_{\eta_n}(\tau')(\omega) < C\right\}$, we have

$$\mathbb{P}(\sup_t |H^{\eta_n,i}(t)| \le C\sqrt{\eta_n}) = \mathbb{P}(\sup_t |Z^{(i)}_{\eta_n}(t)| \le C) \le 1-\delta, \forall n > N,$$

which leads to a contradiction with B.2.3. Our assumption does not hold.

□

### A.3.2 Proof of Proposition 1.5.2

*Proof.* Our analysis is based on approximating $z^{(1)}_{\eta,k}$ by its continuous approximation $Z^{(1)}_{\eta}(t)$, which is normal distributed at time $t$. By simple manipulation, we have

$$\mathbb{P}\left((h^{(2)}_{\eta,N_1})^2 \le 1-\delta^2\right) = \mathbb{P}\left((z^{(2)}_{\eta,N_1})^2 \le \eta^{-1}(1-\delta^2)\right) \ge \mathbb{P}(|z^{(1)}_{\eta,N_1}| \ge \eta^{-\frac{1}{2}}\delta).$$

We then prove $P\left(\left|z^{(1)}_{\eta,N_1}\right| \ge \eta^{-\frac{1}{2}}\delta\right) \ge 1-\nu$. At time t, $z^{(1)}_{\eta,k}$ approximates to a normal distribution with mean $0$ and variance $\frac{\beta_{12}^2}{2(\lambda_1-\lambda_2)}\left[\exp\left(2(\lambda_1-\lambda_2)\eta N_1\right)-1\right]$. Therefore, let $\Phi(x)$ be the CDF of $N(0,1)$, we have

$$\mathbb{P}\left(\frac{\left|z^{(1)}_{\eta,N_1}\right|}{\sqrt{\frac{\beta_{12}^2}{2(\lambda_1-\lambda_2)}\cdot\left[\exp\left(2(\lambda_1-\lambda_2)\eta N_1\right)-1\right]}} \ge \Phi^{-1}\left(\frac{1+\nu}{2}\right)\right) \approx 1-\nu,$$

which requires

$$\eta^{-\frac{1}{2}}\delta \le \Phi^{-1}\left(\frac{1+\nu}{2}\right)\cdot\sqrt{\frac{\beta_{12}^2}{2(\lambda_1-\lambda_2)}\cdot\left[\exp\left(2(\lambda_1-\lambda_2)\eta N_1\right)-1\right]}.$$

Solving the above inequality, we get

$$N_1 = \frac{\eta^{-1}}{2(\lambda_1 - \lambda_2)} \log \left( \frac{2\eta^{-1}\delta^2(\lambda_1 - \lambda_2)}{\Phi^{-1}\left(\frac{1+\nu}{2}\right)^2 \beta_{12}^2} + 1 \right).$$

$\square$

### A.3.3 Proof of Proposition 1.5.3

*Proof.* After Phase I, we restart our counter, i.e., $h_{\eta,0}^{(1)} = \delta$. By (1.4.7) and $h_{\eta,N_2}^{(1)}$ approximating to the process $H^{(1)}(\eta N_2)$, we obtain

$$\begin{aligned}\left(h_{\eta,N_2}^{(1)}(t)\right)^2 = \left(H^{(1)}(\eta N_2)\right)^2 &= \left(\sum_{j=1}^{2d}\left(\left(H^{(j)}(0)\right)^2 \exp\left(2\lambda_j \eta N_2\right)\right)\right)^{-1} \left(H^{(1)}(0)\right)^2 \exp\left(2\lambda_1 \eta N_2\right) \\ &\geq \left(\delta^2 \exp(2\lambda_1\eta N_2) + (1-\delta^2)\exp(2\lambda_2\eta N_2)\right)^{-1} \delta^2 \exp(2\lambda_1 \eta N_2),\end{aligned}$$

which requires

$$\left(\delta^2 \exp(2\lambda_1\eta N_2) + (1-\delta^2)\exp(2\lambda_2\eta N_2)\right)^{-1} \delta^2 \exp(2\lambda_1 \eta N_2) \geq \eta^{-1}(1-\delta^2).$$

Solving the above inequality, we get

$$N_2 = \frac{\eta^{-1}}{2(\lambda_1 - \lambda_2)} \log \frac{1-\delta^2}{\delta^2}\ .$$

$\square$

### A.3.4 Proof of Theorem 1.5.4

*Proof.* For $i = 2, ..., 2d$, we compute the infinitesimal conditional expectation and variance,

$$
\begin{aligned}
&\frac{d}{dt}\mathbb{E}Z^{(i)}_{\eta_n}(t)\big|_{t=t_0} \\
=&\eta^{-1}\mathbb{E}\left[Z^{(i)}_{\eta_n}(t_0+\eta) - Z^{(i)}_{\eta_n}(t_0)\big|H^{\eta}(t_0) = h\right] \\
=&\eta^{-1/2}h_i\sum_{j=1}^{2d}(\lambda_i - \lambda_j)\,h_j^2 + O(\eta) = Z^{(i)}\,(\lambda_i - \lambda_1) + o(1), \\
&\frac{d}{dt}\mathbb{E}\left(Z^{(i)}_{\eta_n}(t) - Z^{(i)}_{\eta_n}(t_0)\right)^2\big|_{t=t_0} \\
=&\eta^{-1}\mathbb{E}\left[\left(Z^{(i)}_{\eta_n}(t_0+\eta) - Z^{(i)}_{\eta_n}(t_0)\right)^2\big|H^{\eta}(t_0) = h\right] \\
=&\eta^{-2}\mathbb{E}\left[\eta^2(\hat{\Lambda}h - h^\top\hat{\Lambda}hh)(\hat{\Lambda}h - h^\top\hat{\Lambda}hh)^\top\right]_{i,i} + O(\eta) \\
=&\mathbb{E}(e_i^\top\hat{\Lambda}e_1e_1^\top\hat{\Lambda}^\top e_i) + o(1) = \frac{1}{4}\left(\gamma_i\omega_1 + \gamma_1\omega_i - 2\,\mathrm{sign}(i-d-1/2)\alpha_{i1}\right) + o(1).
\end{aligned}
$$

Following similar lines to the proof of Theorem 1.5.1, by Section 4 of Chapter 7 in [17], we have for each $k = 2, ..., 2d$, if $Z^{(i)}(0) = \eta^{-1/2}h^{(i)}_{\eta,0}$ as $\eta \to 0^+$, then the stochastic process $\eta^{-1/2}h^{(k)}_{\eta,\lfloor t\eta^{-1}\rfloor}$ weakly converges to the solution of the stochastic differential equation (1.5.5). □

### A.3.5 Proof of Proposition 1.5.5

*Proof.* Since we restart our counter, we have $\sum_{i=2}^{2d}(z^{(i)}_{\eta,0})^2 = \eta^{-1}\delta^2$. Since $z^{(i)}_{\eta,k}$ approximates to $Z^{(i)}(\eta k)$ and its second moment:

$$
\mathbb{E}\left(Z^{(i)}(t)\right)^2 = \frac{\beta_{i1}^2}{2(\lambda_1-\lambda_i)} + \left(\left(Z^{(i)}(0)\right)^2 - \frac{\beta_{i1}^2}{2(\lambda_1-\lambda_i)}\right)\exp\left[-2(\lambda_1-\lambda_i)t\right], \quad \text{for } i \neq 1,
$$

we use the Markov inequality:

$$
\begin{aligned}
&\mathbb{P}\left(\sum_{i=2}^{2d}\left(h_{\eta,N_3}^{(i)}\right)^2>\epsilon\right)\\
\leq&\frac{\mathbb{E}\left(\sum_{i=2}^{2d}\left(h_{\eta,N_3}^{(i)}\right)^2\right)}{\epsilon}=\frac{\mathbb{E}\left(\sum_{i=2}^{2d}\left(z_{\eta,N_3}^{(i)}\right)^2\right)}{\eta^{-1}\epsilon}\\
=&\frac{1}{\eta^{-1}\epsilon}\sum_{i=2}^{2d}\frac{\beta_{i1}^2}{2(\lambda_1-\lambda_i)}\Big(1-\exp\big(-2(\lambda_1-\lambda_i)\eta N_3\big)\Big)\\
&\qquad+\left(z_{\eta,0}^{(i)}\right)^2\exp\left[-2(\lambda_1-\lambda_i)\eta N_3\right]\\
\leq&\frac{1}{\eta^{-1}\epsilon}\Big(\frac{d\max\limits_{2\leq i\leq d}(\beta_{i1}^2)}{2(\lambda_1-\lambda_2)}\Big(1-\exp\big(-2(\lambda_1-\lambda_d)\eta N_3\big)\Big)\\
&\qquad+\frac{d\max\limits_{d+1\leq i\leq 2d}(\beta_{i1}^2)}{2(\lambda_1+\lambda_d)}\Big(1-\exp\big(-4\lambda_1\eta N_3\big)\Big)+\delta^2\exp\left[-2(\lambda_1-\lambda_2)\eta N_3\right]\Big)\\
\leq&\frac{1}{\eta^{-1}\epsilon}\left(\frac{d\max\limits_{1\leq i\leq d}(\beta_{i1}^2)}{(\lambda_1-\lambda_2)}+\delta^2\exp\left[-2(\lambda_1-\lambda_2)\eta N_3\right]\right).
\end{aligned}
$$

To guarantee $\frac{1}{\eta^{-1}\epsilon}\left(\frac{d\max_{1\leq i\leq d}(\beta_{i1}^2)}{(\lambda_1-\lambda_2)}+\delta^2\exp\left[-2(\lambda_1-\lambda_2)\eta N_3\right]\right)\leq\frac{1}{4}$, we get:

$$
N_3\geq\frac{\eta^{-1}}{2(\lambda_1-\lambda_2)}\log\left(\frac{4(\lambda_1-\lambda_2)\delta^2}{(\lambda_1-\lambda_2)\epsilon\eta^{-1}-4d\max\limits_{1\leq i\leq d}\beta_{i1}^2}\right).
$$

□

### A.3.6 Proof of Corollary 1.5.6

*Proof.* First, we prove that $\|u_{\eta,k}-\hat{u}\|_2^2+\|v_{\eta,k}-\hat{v}\|_2^2$ can be bounded by $3\sum_{i=2}^{2d}\left(h_{\eta,k}^{(i)}\right)^2$, when it is near the optima. Recall that $h_{\eta,k}=\frac{1}{\sqrt{2}}P^\top(u_{\eta,k}^\top\,v_{\eta,k}^\top)^\top$ and $e_1=\hat{h}=\frac{1}{\sqrt{2}}\mathbb{P}(\hat{u}^\top\,\hat{v}^\top)^\top$. Our analysis has shown that when $k$ is large enough, the SGD iterates near the optima. Then we have

$$
\begin{aligned}
\|u_{\eta,k}-\hat{u}\|_2^2+\|v_{\eta,k}-\hat{v}\|_2^2 &= 4-2\langle u_{\eta,k},\hat{u}\rangle-2\langle v_{\eta,k},\hat{v}\rangle \\
&= 4-4h_{\eta,k}^1 = 4-4\sqrt{1-\sum\nolimits_{i=2}^{2d}\left(h_{\eta,k}^{(i)}\right)^2} \\
&= \frac{16\sum_{i=2}^{2d}\left(h_{\eta,k}^{(i)}\right)^2}{4+4\sqrt{1-\sum_{i=2}^{2d}\left(h_{\eta,k}^{(i)}\right)^2}} \le 3\sum\nolimits_{i=2}^{2d}\left(h_{\eta,k}^{(i)}\right)^2, \qquad \text{(A.3.4)}
\end{aligned}
$$

where the last inequality holds since $k$ is large enough such that $\sum_{i=2}^{2d}\left(h_{\eta,k}^{(i)}\right)^2$ is sufficiently small. By Propositions 1.5.2, 1.5.3, and 1.5.5, the total iteration number is

$$
N = N_1+N_2+N_3. \qquad \text{(A.3.5)}
$$

To explicititly bound $N$ in (B.2.10) in terms of sample size n, we consider

$$
N_1 = \frac{\eta^{-1}}{2(\lambda_1-\lambda_2)}\log\left(\frac{2\eta^{-1}\delta^2(\lambda_1-\lambda_2)}{\Phi^{-1}\left(\frac{1+\nu}{2}\right)^2\beta_{12}^2}+1\right), \qquad \text{(A.3.6)}
$$

$$
N_2 = \frac{\eta^{-1}}{2(\lambda_1-\lambda_2)}\log\frac{1-\delta^2}{\delta^2}, \qquad \text{(A.3.7)}
$$

$$
N_3 = \frac{\eta^{-1}}{2(\lambda_1-\lambda_2)}\log\left(\frac{4(\lambda_1-\lambda_2)\delta^2}{(\lambda_1-\lambda_2)\epsilon\eta^{-1}-4d\max\limits_{1\le i\le d}\beta_{i1}^2}\right). \qquad \text{(A.3.8)}
$$

Given a small enough $\epsilon$, we choose $\eta$ as follow:

$$
\eta \asymp \frac{\epsilon(\lambda_1-\lambda_2)}{d\max_{1\le i\le d}\beta_{i1}^2}. \qquad \text{(A.3.9)}
$$

Combining the above sample complexities (A.3.6), (A.3.7), (A.3.8), and (A.3.9), we get

$$
N = O\left[\frac{d}{\epsilon(\lambda_1-\lambda_2)^2}\log\left(\frac{d}{\epsilon}\right)\right]. \qquad \text{(A.3.10)}
$$

By Proposition 1.5.5 with (A.3.4), after at most $N$ iterations, we have

$$\|u_{\eta,n} - \hat{u}\|_2^2 + \|v_{\eta,n} - \hat{v}\|_2^2 \leq 3\|h_{\eta,n} - \hat{h}\|_2^2 \leq 3\epsilon,$$

with probability at least $\frac{3}{4}$. $\square$

# APPENDIX B

# SUPPLEMENTARY MATERIALS IN CHAPTER 2

## B.1 Detailed Proofs in Section 2.3

### B.1.1 Proof of Theorem 2.3.4

*Proof.* Remind that the eigendecomposition of $\tilde{A}$ is $(\Lambda^B)^{-\frac{1}{2}}O^{B\top}AO^B(\Lambda^B)^{-\frac{1}{2}} = O^{\tilde{A}}\Lambda^{\tilde{A}}(O^{\tilde{A}})^\top$. Given the eigendecomposition of $B$ is $B = O^B\Lambda^B(O^B)^\top$, we can write $B^{-1}$ as

$$B^{-1} = O^B(\Lambda^B)^{-1}(O^B)^\top.$$

We denote $\tilde{X}$ as $\tilde{X} = O^{\tilde{A}}_{:,\mathcal{I}}$ for some $\mathcal{I} \subseteq [d]$ with $|\mathcal{I}| = r$. For $X = (B^{-1/2}O^{\tilde{A}}_{:,\mathcal{I}}) \cdot \Psi$, where $\Psi \in \mathcal{G}$. It is easy to see that $\nabla_Y \mathcal{L}(X, Y) = 0$. Ignore the constant 2 in the gradient $\nabla_X \mathcal{L}(X, Y)$ for convenience, we have,

$$\begin{aligned}
\nabla_X \mathcal{L}(X, Y) &= -(I_d - BXX^\top)AX = -(I_d - BB^{-1/2}O^{\tilde{A}}_{:,\mathcal{I}}(O^{\tilde{A}}_{:,\mathcal{I}})^\top B^{-1/2})AB^{-1/2}O^{\tilde{A}}_{:,\mathcal{I}} \\
&= -AB^{-1/2}O^{\tilde{A}}_{:,\mathcal{I}} + B^{1/2}O^{\tilde{A}}_{:,\mathcal{I}}(O^{\tilde{A}}_{:,\mathcal{I}})^\top O^{\tilde{A}}\Lambda^{\tilde{A}}(O^{\tilde{A}})^\top O^{\tilde{A}}_{:,\mathcal{I}} \\
&= -B^{1/2}O^{\tilde{A}}\Lambda^{\tilde{A}}(O^{\tilde{A}})^\top O^{\tilde{A}}_{:,\mathcal{I}} + B^{1/2}O^{\tilde{A}}_{:,\mathcal{I}}\Lambda^{\tilde{A}}_{\mathcal{I},\mathcal{I}} \\
&= -B^{1/2}O^{\tilde{A}}\Lambda^{\tilde{A}}_{:\mathcal{I}} + B^{1/2}O^{\tilde{A}}_{:,\mathcal{I}}\Lambda^{\tilde{A}}_{\mathcal{I},\mathcal{I}} = 0.
\end{aligned}$$

Next we show that if $X$ is not as specified, then $\nabla_X \mathcal{L}(X, Y) \neq 0$. We only need to show that if $\tilde{X} = [O^{\tilde{A}}_{:,\mathcal{S}},\ \phi]\Psi$, where $\mathcal{S} \subseteq [d]$ with $|\mathcal{S}| = r - 1$ and $\phi = c_1 O^{\tilde{A}}_{:,i} + c_2 O^{\tilde{A}}_{:,j}$ with $i, j \notin \mathcal{S}$, $i \neq j$, $c_1^2 + c_2^2 = 1$, and $c_1, c_2 \neq 0$, then we have $\nabla_X \mathcal{L}(X, Y) \neq 0$. The general scenario can be induced from this basic setting. It is easy to see that such an $X = B^{-1/2}\tilde{X}$

satisfies the constraint,

$$X^\top BX = \Psi^\top [O^{\tilde{A}}_{:,\mathcal{S}},\ \phi]^\top B^{-1/2} B B^{-1/2} [O^{\tilde{A}}_{:,\mathcal{S}},\ \phi]\Psi = \Psi^\top \begin{bmatrix} I_{r-1} & 0_{(r-1)\times 1} \\ 0_{1\times(r-1)} & \phi^\top\phi \end{bmatrix} \Psi = I_r,$$

where the last equality follow from $\phi^\top\phi = c_1^2 + c_2^2 = 1$.

Plugging such an $X$ into the gradient, we have

$$\begin{aligned}
&\nabla_X \mathcal{L}(X,Y) \\
&= -(I_d - BXX^\top)AX \\
&= -(I_d - BB^{-1/2}[O^{\tilde{A}}_{:,\mathcal{S}},\ \phi][O^{\tilde{A}}_{:,\mathcal{S}},\ \phi]^\top B^{-1/2})AB^{-1/2}[O^{\tilde{A}}_{:,\mathcal{S}},\ \phi]\Psi \\
&= -B^{1/2}(O^{\tilde{A}}_{:,\mathcal{S}^\perp}(O^{\tilde{A}}_{:,\mathcal{S}^\perp})^\top - \phi\phi^\top)O^{\tilde{A}}\Lambda^{\tilde{A}}[(I_d)_{\mathcal{S}},\ c_1e_i + c_2e_j]\Psi \\
&= -B^{1/2}[0_{d\times(r-1)},\ O^{\tilde{A}}_{:,\mathcal{S}^\perp}\Lambda^{\tilde{A}}_{\mathcal{S}^\perp,:}(c_1e_i + c_2e_j)]\Psi + [0_{d\times(r-1)},\ \phi(c_1^2\lambda_i^{\tilde{A}} + c_2^2\lambda_j^{\tilde{A}})]\Psi \\
&= -B^{1/2}[0_{d\times(r-1)},\ c_1c_2^2(\lambda_i^{\tilde{A}} + \lambda_j^{\tilde{A}})O^{\tilde{A}}_{:,i} + c_2c_1^2(\lambda_j^{\tilde{A}} - \lambda_i^{\tilde{A}})O^{\tilde{A}}_{j,j}]\Psi \neq 0,
\end{aligned}$$

where the last $\neq$ is from $c_1, c_2 \neq 0$, $c_1^2 + c_2^2 = 1$, $\lambda_j^{\tilde{A}} \neq \lambda_j^{\tilde{A}}$ for $i \neq j$. $\square$

### B.1.2 Proof of Theorem 2.3.5

*Proof.* We have the Hessian of $\mathcal{L}(X,Y)$ on $X$ with $Y = \mathcal{D}(X)$ as

$$H_X = 2\,\mathrm{sym}\left(I_r \otimes ((BXX^\top - I_d)A) + (X^\top AX) \otimes B + (AX) \boxtimes (BX)\right) \quad \text{(B.1.1)}$$

where $\mathrm{sym}(M) = M + M^\top$, $\otimes$ is the Kronecker product, and for $U \in \mathbb{R}^{d\times r}$ and $V \in \mathbb{R}^{m\times k}$, $U \boxtimes V \in \mathbb{R}^{dk\times mr}$ is defined as

$$U \boxtimes V = \begin{bmatrix} U_{:,1}V_{:,1}^\top & U_{:,2}V_{:,1}^\top & \cdots & U_{:,r}V_{:,1}^\top \\ U_{:,1}V_{:,2}^\top & U_{:,2}V_{:,2}^\top & \cdots & U_{:,r}V_{:,2}^\top \\ \vdots & \vdots & \ddots & \vdots \\ U_{:,1}V_{:,k}^\top & U_{:,2}V_{:,k}^\top & \cdots & U_{:,r}V_{:,k}^\top \end{bmatrix}.$$

To determine whether a stationary point is an unstable stationary or a minimax global optimum, we consider its Hessian. We start with checking that $\mathcal{S} = [r]$ corresponds to the global optimum, $X = B^{-1/2} O^{\tilde{A}}_{:,[r]} \Psi$. Without loss of generality, we set $\Psi = I_r$. We only need to check that for any vector $v = [v_1^\top, \ldots, v_r^\top]^\top \in \mathbb{R}^{nr}$ with $v_i \in \mathbb{R}^n$ denoting the $i$-th block of $v$, which satisfies

$$v_i = c_{j_i} B^{-1/2} O^{\tilde{A}}_{:,j_i} \text{ for any } j_i \in [d] \text{ and a real constant } c_j,$$

such that $\|v\|_2 = 1$, then we have $v^\top H_X v \geq 0$. The general case is only a linear combination of such $v$'s. Specifically, for $X = O^{\tilde{A}}_{:,[r]}$, we have

$$\begin{aligned}
v^\top H_X v &= -v^\top \operatorname{sym}\left(I_r \otimes ((I_d - BXX^\top)A) - (X^\top A X) \otimes B - (AX) \boxtimes (BX)\right) v \\
&= -v^\top \operatorname{sym}\Big(I_r \otimes ((I_d - B^{1/2} O^{\tilde{A}}_{:,[r]} O^{\tilde{A}\top}_{:,[r]} B^{-1/2})A) - (O^{\tilde{A}\top}_{:,[r]} B^{-1/2} A B^{-1/2} O^{\tilde{A}}_{:,[r]}) \otimes B \\
&\qquad\qquad - (AB^{-1/2} O^{\tilde{A}}_{:,[r]}) \boxtimes (B^{1/2} O^{\tilde{A}}_{:,[r]})\Big) v \\
&= -v^\top \operatorname{sym}\Big(I_r \otimes (B^{1/2} O^{\tilde{A}}_{:,[d]\backslash[r]} O^{\tilde{A}\top}_{:,[d]\backslash[r]} B^{-1/2} A) \\
&\qquad\qquad - \Lambda^{\tilde{A}}_{:,[r]} \otimes B - (B^{1/2} O^{\tilde{A}} \Lambda^{\tilde{A}}_{:,[r]}) \boxtimes (B^{1/2} O^{\tilde{A}}_{:,[r]})\Big) v \\
&= -2 \sum_{i=1}^{r} c_{j_i}^2 O^{\tilde{A}\top}_{:,j_i} O^{\tilde{A}}_{:,[d]\backslash[r]} \Lambda^{\tilde{A}}_{[d]\backslash[r],:} O^{\tilde{A}\top} O^{\tilde{A}}_{:,j_i} + 2 \sum_{i=1}^{r} c_{j_i}^2 \lambda_i^{\tilde{A}} \\
&\qquad\qquad + 2 \sum_{i=1}^{r} \sum_{k=1}^{r} c_{j_i} c_{j_k} e_{j_i}^\top \Lambda^{\tilde{A}}_{:,k} O^{\tilde{A}\top}_{:,i} O^{\tilde{A}}_{j_k} \\
&\geq 0 + 2 \sum_{i=1}^{r} c_{j_i}^2 \lambda_i^{\tilde{A}} + 2 \sum_{i=1}^{r} \sum_{k=1}^{r} c_{j_i} c_{j_k} \lambda_{j_i}^{\tilde{A}} = 0,
\end{aligned}$$

where the last inequality is obtained by taking $j_k \in [r]$, $i = j_k$, and $k = j_i$ in the last term, and the last equality is obtained by setting $c_{j_k} = -c_{j_i}$ when $j_i = k$, which implies that the restricted strongly convex property at $X$ holds.

For any other $\mathcal{I} \neq [r]$, we only need to show that the largest eigenvalue of $\nabla^2 \mathcal{L}$ is positive and the smallest eigenvalue of $\nabla^2 \mathcal{L}$ is negative, which implies that such a stationary point is unstable. Using the same construction as above, we have

$$\lambda_{\min}(H_X) \leq -v^\top \operatorname{sym}\left(I_r \otimes ((I_d - BXX^\top)A) - (X^\top AX) \otimes B - (AX) \boxtimes (BX)\right) v$$
$$= -v^\top \operatorname{sym}\Big(I_r \otimes (B^{1/2} O^{\tilde{A}}_{:,\mathcal{I}^\perp} O^{\tilde{A}\top}_{:,\mathcal{I}^\perp} B^{-1/2} A)$$
$$- \Lambda^{\tilde{A}}_{:,\mathcal{I}} \otimes B - (B^{1/2} O^{\tilde{A}} \Lambda^{\tilde{A}}_{:,\mathcal{I}}) \boxtimes (B^{1/2} O^{\tilde{A}}_{:,\mathcal{I}})\Big) v$$
$$= -2 \sum_{i \in \mathcal{I}} c_{j_i}^2 O^{\tilde{A}\top}_{:,j_i} O^{\tilde{A}}_{:,\mathcal{I}^\perp} \Lambda^{\tilde{A}}_{\mathcal{I}^\perp,:} O^{\tilde{A}\top} O^{\tilde{A}}_{:,j_i} + 2 \sum_{i \in \mathcal{I}} c_{j_i}^2 \lambda_i^{\tilde{A}} + 2 \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{I}} c_{j_i} c_{j_k} e_{j_i}^\top \Lambda^{\tilde{A}}_{:,k} O^{\tilde{A}\top}_{:,i} O^{\tilde{A}}_{j_k}$$
$$\overset{(i)}{=} 2 c_{j_r}^2 (\lambda^{\tilde{A}}_{\max \mathcal{I}} - \lambda^{\tilde{A}}_{\min \mathcal{I}^\perp}),$$

where $(i)$ is from setting $c_{j_i} = 0$ for all $j_i \in \mathcal{I}^\perp$ except $j_r$, and $c_{j_r} = 1/\|B^{-1/2} O^{\tilde{A}}_{:,\min \mathcal{I}^\perp}\|_2$.

On the other hand, we have

$$\lambda_{\max}(H_X) \geq v^\top H_X v$$
$$= -2 \sum_{i \in \mathcal{I}} c_{j_i}^2 O^{\tilde{A}\top}_{:,j_i} O^{\tilde{A}}_{:,\mathcal{I}^\perp} \Lambda^{\tilde{A}}_{\mathcal{I}^\perp,:} O^{\tilde{A}\top} O^{\tilde{A}}_{:,j_i} + 2 \sum_{i \in \mathcal{I}} c_{j_i}^2 \lambda_i^{\tilde{A}} + 2 \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{I}} c_{j_i} c_{j_k} e_{j_i}^\top \Lambda^{\tilde{A}}_{:,k} O^{\tilde{A}\top}_{:,i} O^{\tilde{A}}_{j_k}$$
$$\overset{(i)}{=} 2 c_{j_1}^2 \lambda^{\tilde{A}}_{\min \mathcal{I}} + c_{j_1}^2 \lambda^{\tilde{A}}_{\min \mathcal{I}} = 4 c_{j_1}^2 \lambda^{\tilde{A}}_{\min \mathcal{I}},$$

where $(i)$ is from setting $c_{j_i} = 0$ for all $j_i \in \mathcal{I}$ except $j_1$, and $c_{j_1} = 1/\|B^{-1/2} O^{\tilde{A}}_{:,\min \mathcal{I}}\|_2$.

$\square$

Then we will discuss the assumption 2.3.1: When $B$ **is Singular**, we assume $\operatorname{rank}(B) = m < d$ and $\operatorname{rank}(A) = d$. Note that we require $m \geq r$; Otherwise, the feasible region of (2.1.3) becomes $\mathcal{T}_B = \emptyset$.

Before we proceed with our analysis, we first exclude an ill-defined case, where the objective function of (2.1.3) is unbounded from above. The following proposition shows the sufficient and necessary condition of the existence of the global optima of (2.1.3).

**Proposition B.1.1.** Given a full rank symmetric matrix $A \in \mathbb{R}^{d \times d}$ and a positive semidefinite matrix $B \in \mathbb{R}^{d \times d}$, the optimal solution of (2.1.3) exists if and only if for all $v \in$ Null$(B)$, one of the following two condition holds: (1) $v^\top A v < 0$; (2) $v^\top A v = 0$ and $u^\top A v = 0, \forall u \in$ Col$(B)$.

*Proof.* We decompose $X = X_B + X_{B^\perp}$, where $X_B = [u_1, ..., u_r]$ with $u_i \in \text{Col}(B)$ and each column of $X_{B^\perp} = [v_1, ..., v_r]$ with $v_i \in \text{Null}(B)$. Note such decomposition is unique. Then (2.1.3) becomes

$$\min -\sum_{i=1}^{r}(u_i^\top A u_i) - 2\sum_{i=1}^{r}(u_i^\top A v_i) - \sum_{i=1}^{r}(v_i^\top A v_i) \quad \text{s.t.} \quad X_B^\top B X_B = I_r. \qquad \text{(B.1.2)}$$

If (B.1.2) has an optimal solution, we have $v^\top A v \leq 0$, for all $v \in \text{Null}(B)$; otherwise, fixing the feasible $X_B$, we use $X_B = [\lambda v, ..., \lambda v]$ and increase $\lambda$, then there is no lower bound of objective value. Further, given a vector $v \in \text{Null}(B)$ with $v^\top A v = 0$, $u^\top A v = 0$ must hold for all $u \in \text{Col}(B)$; otherwise, W.L.O.G, we assume that $u_1 \in \text{Col}(B)$ $u_1^\top A v > 0$, we can construct a feasible $X_B = \mu[u_1, ..., u_r]$, where $\mu$ is a normalization constant such that $\mu^2 u_1^\top B u_1 = 1$. Then constructing $X_{B^\perp} = \lambda[v, 0, ...0]$., if we increase $\lambda$, there is no lower bound the objective value. Therefore, for a vector $v \in \text{Null}(B)$, either $v^\top A v = 0$, or $u^\top A v = 0$ and $v^\top A v = 0$ hold. □

Throughout our following analysis, we exclude the ill-defined case.

The idea of characterizing all the equilibria is analogous to the nonsingular case, but much more involved. Since B is singular, we need to use general inverses. For notationally convenience, we use block matrices in our analysis. We consider the eigenvalue decomposition of $B$ as follows:

$$B = \underbrace{\begin{bmatrix} O_{11}^B & O_{12}^B \\ O_{21}^B & O_{22}^B \end{bmatrix}}_{O^B} \underbrace{\begin{bmatrix} \Lambda_{11}^B & 0 \\ 0 & 0 \end{bmatrix}}_{\Lambda^B} \underbrace{\begin{bmatrix} O_{11}^{B\top} & O_{21}^{B\top} \\ O_{12}^{B\top} & O_{22}^{B\top} \end{bmatrix}}_{O^{B\top}},$$

where $O_{11}^B \in \mathbb{R}^{m \times m}$, $O_{22}^B \in \mathbb{R}^{(d-m)\times(d-m)}$, and $\Lambda_{11}^B = \text{diag}(\lambda_1, ..., \lambda_m)$ with $\lambda_1 \geq \cdots \geq$

$\lambda_m > 0$ . We then left multiply $O^{B\top}$ and right multiply $O^B$ to $A$:

$$O^{B\top} A O^B =: W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix},$$

where $W_{11} \in \mathbb{R}^{m\times m}, W_{22} \in \mathbb{R}^{(d-m)\times(d-m)}$. Here, we assume $W_{22}$ is nonsingular (guaranteed in the well-defined case). Then we construct a general inverse of $\Lambda^B$. Specifically, given an arbitrary positive definite matrix $P \in \mathbb{R}^{(d-m)\times(d-m)}$, we define $\Lambda^{B\dagger}(P)$ as

$$\Lambda^{B\dagger}(P) := \begin{bmatrix} (\Lambda_{11}^B)^{-1} & 0 \\ 0 & P \end{bmatrix}.$$

Note $\Lambda^{B\dagger}(P)$ is invertible and depends on $P$. Recall the primal variable $X$ at the equilibrium of $\mathcal{L}(X, Y)$ satisfies

$$AX = BX \cdot X^\top AX \ \text{ and } \ X^\top BX = I_r. \tag{B.1.3}$$

For notational simplicity, we define

$$V(P) := \left(\Lambda^{B\dagger}(P)\right)^{-\frac{1}{2}} O^{B\top} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2(P) \end{bmatrix}, \tag{B.1.4}$$

where $V_1,\ X_1 \in \mathbb{R}^{m\times r}$, and $V_2(P),\ X_2 \in \mathbb{R}^{(d-m)\times r}$. Note that $V_1$ does not depend on $P$. From (B.1.4) we have

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = O^B \left(\Lambda^{B\dagger}(P)\right)^{\frac{1}{2}} \begin{bmatrix} V_1 \\ V_2(P) \end{bmatrix}. \tag{B.1.5}$$

Combining (B.1.5) and (B.1.3) we get the following equation system:

$$\begin{cases} \tilde{A}(P)V(P) = \begin{bmatrix} V_1 \\ 0 \end{bmatrix} V(P)^\top \tilde{A}(P)V(P), & \text{(B.1.6a)} \\ V(P)^\top \mathrm{diag}(I_m, 0)V(P) = I_r, & \text{(B.1.6b)} \end{cases}$$

where $\tilde{A}(P) = (\Lambda^{B\dagger}(P))^{\frac{1}{2}} W (\Lambda^{B\dagger}(P))^{\frac{1}{2}}$. The invertibility of $\Lambda^{B\dagger}(P)$ ensures that solving (B.1.3) is equivalent to doing the transformation (B.1.5) to the solution of (B.1.6). We then denote

$$\hat{A} = (\Lambda_{11}^B)^{-\frac{1}{2}} \left( W_{11} - W_{12} W_{22}^{-1} W_{21} \right) (\Lambda_{11}^B)^{-\frac{1}{2}}$$

and consider its eigenvalue decomposition as $\hat{A} = O^{\hat{A}} \Lambda^{\hat{A}} O^{\hat{A}\top}$. The following theorem characterizes all the equilibria of $\mathcal{L}(X, Y)$ with a singular $B$.

**Theorem B.1.2.** Given a full rank symmetric matrix $A \in \mathbb{R}^{d\times d}$ and a positive semidefinite matrix $B \in \mathbb{R}^{d\times d}$ with $\mathrm{rank}(B) = m < d$, satisfying the well-defined condition in Proposition B.1.1,$(X, \mathcal{D}(X))$ is an equilibrium of $\mathcal{L}(X, Y)$ if and only if $X$ can be represented as

$$X = O^B \begin{bmatrix} (\Lambda_{11}^B)^{-\frac{1}{2}} \cdot O^{\hat{A}}_{:,\mathcal{I}} \\ -W_{22}^{-1} W_{12}^\top (\Lambda_{11}^B)^{-\frac{1}{2}} O^{\hat{A}}_{:,\mathcal{I}} \end{bmatrix} \cdot \Psi,$$

where $\Psi \in \mathcal{G}$ and $\mathcal{I} \in \mathcal{X}_m$ is the column index set.

*Proof.* By definition, we have

$$\begin{cases} AX = BX \cdot Y \\ X^\top BX = I_r \end{cases} \Longrightarrow \begin{cases} AX = BX \cdot X^\top AX \\ X^\top BX = I_r \end{cases}, \qquad \text{(B.1.7)}$$

We define $V(P) := \left(\Lambda^{B\dagger}(P)\right)^{-\frac{1}{2}} O^{B\top} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2(P) \end{bmatrix}$, where $V_1,\ X_1 \in \mathbb{R}^{m\times r}$, and $V_2(P),\ X_2 \in \mathbb{R}^{(d-m)\times r}$. Note that $V_1$ does not depend on $P$. By (B.1.7) and replacing $I_d$

with $O^B O^{B\top}$ and $\Lambda^{B\dagger}(P)^{\frac{1}{2}}\Lambda^{B\dagger}(P)^{-\frac{1}{2}}$, we have

$$\begin{cases} \tilde{A}(P)V(P) = \begin{bmatrix} V_1 \\ 0 \end{bmatrix} V(P)^\top \tilde{A}(P)V(P), & \text{(B.1.8a)} \\ V(P)^\top \mathrm{diag}(I_m, 0)V(P) = I_r, & \text{(B.1.8b)} \end{cases}$$

where $\tilde{A}(P) = (\Lambda^{B\dagger}(P))^{\frac{1}{2}} W (\Lambda^{B\dagger}(P))^{\frac{1}{2}}$. Simplifying (B.1.8a), we obtain

$$\begin{cases} W_{22}^{-1}W_{21}(\Lambda_{11}^B)^{-\frac{1}{2}}V_1 = P^{\frac{1}{2}}V_2(P) \\ V_1V_1^\top(\Lambda_{11}^B)^{-\frac{1}{2}}(W_{11} - W_{12}W_{22}^{-1}W_{21})(\Lambda_{11}^B)^{-\frac{1}{2}} \end{cases}$$

Let $\hat{A} = (\Lambda_{11}^B)^{-\frac{1}{2}}\left(W_{11} - W_{12}W_{22}^{-1}W_{21}\right)(\Lambda_{11}^B)^{-\frac{1}{2}}$. Then, by (B.1.8), we obtain the following equations:

$$\begin{cases} \hat{A}V_1 = V_1V_1^\top \hat{A}V_1, & \text{(B.1.9a)} \\ V_1^\top V_1 = I_r, & \text{(B.1.9b)} \end{cases}$$

Note (B.1.9) are the KKT conditions of the following problem:

$$V_1^* = \operatorname*{argmin}_{V_1 \in \mathbb{R}^{m\times r}} -\mathrm{tr}(V_1^\top \hat{A} V_1) \quad \text{s.t.} \quad V_1^\top V_1 = I_r. \tag{B.1.10}$$

Because (B.1.10) is not a degenerate case, Theorem 2.3.4 can be directly applied to (B.1.10). Then, we get the stable equilibria and unstable equilibria of (B.1.10). Specifically, denote the eigenvalue decomposition of $\hat{A}$ as $\hat{A} = O^{\hat{A}}\Lambda^{\hat{A}}O^{\hat{A}\top}$. Then we know the equilibrium of (B.1.9) can be represented as $V_1 = O^{\hat{A}}_{:,\mathcal{I}} \cdot \Psi$, where $\mathcal{I} \in \Big\{\{i_1, ..., i_r\} : \{i_1, ..., i_r\} \subseteq [m]\Big\}$ and $\Psi \in \mathcal{G}$. Then, we know the primal variable $X$ at an equilibrium of $\mathcal{L}(X, Y)$ satisfies

$$X = O^B \begin{bmatrix} (\Lambda_{11}^B)^{-\frac{1}{2}} \cdot O^{\hat{A}}_{:,\mathcal{I}} \\ -W_{22}^{-1}W_{12}^\top(\Lambda_{11}^B)^{-\frac{1}{2}}O^{\hat{A}}_{:,\mathcal{I}} \end{bmatrix} \cdot \Psi,$$

where $O^{\hat{A}}_{:,\mathcal{I}}$ is an equilibrium for the Lagrangian function of (B.1.10). □

Theorem B.1.2 implies that for the well-defined degenerated case, there are only $\binom{m}{r}$ equilibria unique in the sense of invariant group, since $B$ is rank deficient.

## B.2 Detailed Proofs in Section 2.4

### B.2.1 Proof of Lemma 2.4.2

*Proof.* Denote $k = \lfloor \frac{t}{\eta} \rfloor$, $\Delta(t) = w^{(\eta)}(t+\eta) - w^{(\eta)}(t)$, $\Delta_i$ as the $i$-th component of $\Delta$. For notational simplicity, we may drop $(t)$ if it is clear from the context. By the definition of $w^{\eta}(t)$, we have

$$\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left(\Delta(t)\middle|w^{(\eta)}(t)\right) \\
&= \frac{1}{\eta}\mathbb{E}\left(W^{(k+1)} - W^{(k)}\middle|W^{(k)}\right) \\
&= \frac{1}{\eta}\mathbb{E}\left[\eta\left(\Lambda^B - (\Lambda^B)^{\frac{1}{2}}\hat{\Lambda}_B^{(k)}(\Lambda^B)^{-\frac{1}{2}}W^{(k)}W^{(k)\top}\right)\cdot\tilde{\Lambda}^{(k)}W^{(k)}\middle|W^{(k)}\right] \\
&= \Lambda^A w^{(\eta)}(t) - \left(w^{(\eta)}(t)\right)^\top\left(\Lambda^B\right)^{-\frac{1}{2}}\Lambda^A\left(\Lambda^B\right)^{-\frac{1}{2}}w^{(\eta)}(t)\Lambda^B w^{(\eta)}(t). \qquad \text{(B.2.1)}
\end{aligned}$$

Similarly, we calculate the infinitesimal conditional expectation of $v_{i,1} = \frac{(w_i^{(\eta)})^{\mu_1}}{(w_1^{(\eta)})^{\mu_i}}$ as

$$
\begin{aligned}
&\tfrac{1}{\eta}\mathbb{E}\left(\frac{\left(w_i^{(\eta)}\right)^{\mu_1}}{\left(w_1^{(\eta)}\right)^{\mu_i}}(t+\eta) - \frac{\left(w_i^{(\eta)}\right)^{\mu_1}}{\left(w_1^{(\eta)}\right)^{\mu_i}}(t)\middle|\frac{\left(w_i^{(\eta)}\right)^{\mu_1}}{\left(w_1^{(\eta)}\right)^{\mu_i}}(t)\right)\\
=&\frac{1}{\eta}\mathbb{E}\left(\frac{\left(w_i^{(\eta)}(t)+\Delta_i\right)^{\mu_1}}{\left(w_1^{(\eta)}(t)+\Delta_1\right)^{\mu_i}} - \frac{\left(w_i^{(\eta)}(t)\right)^{\mu_1}}{\left(w_1^{(\eta)}(t)\right)^{\mu_i}}\middle|\frac{\left(w_i^{(\eta)}(t)\right)^{\mu_1}}{\left(w_1^{(\eta)}(t)\right)^{\mu_i}}\right)\\
=&\frac{1}{\eta}\frac{\left(w_i^{(\eta)}(t)\right)^{\mu_1}}{\left(w_1^{(\eta)}(t)\right)^{\mu_i}}\mathbb{E}\left([1+\mu_1\frac{\Delta_i}{w_i^{(\eta)}}+\mathcal{O}(\eta^2)]\cdot[1-\mu_i\frac{\Delta_1}{w_1^{(\eta)}}+\mathcal{O}(\eta^2)]-1\middle|w^{(\eta)}(t)\right)\\
=&\frac{1}{\eta}\frac{\left(w_i^{(\eta)}(t)\right)^{\mu_1}}{\left(w_1^{(\eta)}(t)\right)^{\mu_i}}\left(\frac{\mu_1}{w_i^{(\eta)}}\mathbb{E}(\Delta_i\big|w^{(\eta)}(t)) - \frac{\mu_i}{w_1^{(\eta)}}\mathbb{E}(\Delta_1\big|w^{(\eta)}(t))\right)+\mathcal{O}(\eta)\\
=&\frac{\left(w_i^{(\eta)}(t)\right)^{\mu_1}}{\left(w_1^{(\eta)}(t)\right)^{\mu_i}}\Big[\frac{\mu_1}{w_i^{(\eta)}}\left(-\sum_{k=1}^{d}\frac{\lambda_k}{\mu_k}\left(w_k^{(\eta)}\right)^2\mu_i w_i^{(\eta)}+\lambda_i w_i^{(\eta)}\right)-\\
&\qquad\frac{\mu_i}{w_1^{(\eta)}}\left(-\sum_{k=1}^{d}\frac{\lambda_k}{\mu_k}\left(w_k^{(\eta)}\right)^2\mu_1 w_1^{(\eta)}+\lambda_1 w_1^{(\eta)}\right)\Big]+\mathcal{O}(\eta)\\
=&\frac{\left(w_i^{(\eta)}(t)\right)^{\mu_1}}{\left(w_1^{(\eta)}(t)\right)^{\mu_i}}\mu_1\mu_i\left(\beta_i-\beta_1\right)+\mathcal{O}(\eta),
\end{aligned}
$$

where the third equality holds because of the Taylor expansion, the fourth holds for $\Delta$ is order of $\mathcal{O}(\eta)$ and the last equality holds due to (B.2.1). Then, we calculate the infinitesimal conditional variance. From the update of $W$ in (2.4.4), if $t \in [0,\ T]$ with a finite $T$, then

$w^{(\eta)}(t)$ is bounded with probability 1. Denote $\|w^{(\eta)}(t)\|_2^2 \le D < \infty$. Then we have

$$
\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left[\left(\frac{\left(w_i^{(\eta)}\right)^{\mu_1}}{\left(w_1^{(\eta)}\right)^{\mu_i}}(t+\eta) - \frac{\left(w_i^{(\eta)}\right)^{\mu_1}}{\left(w_1^{(\eta)}\right)^{\mu_i}}(t)\right)^2 \middle| \frac{\left(w_i^{(\eta)}\right)^{\mu_1}}{\left(w_1^{(\eta)}\right)^{\mu_i}}(t)\right] \\
=&\frac{1}{\eta}\frac{\left(w_i^{(\eta)}(t)\right)^{2\mu_1}}{\left(w_1^{(\eta)}(t)\right)^{2\mu_i}}\mathbb{E}\left[\left(\mu_1\frac{\Delta_i}{w_i^{(\eta)}} - \mu_i\frac{\Delta_1}{w_1^{(\eta)}}\right)^2 \middle| w^{(\eta)}(t)\right] + \mathcal{O}(\eta^2) \\
\le&\frac{2}{\eta}\frac{\left(w_i^{(\eta)}(t)\right)^{2\mu_1}}{\left(w_1^{(\eta)}(t)\right)^{2\mu_i}}\mathbb{E}\left[\left(\frac{\mu_1}{w_i^{(\eta)}}\right)^2\Delta_i^2 + \left(\frac{\mu_i}{w_1^{(\eta)}}\right)^2\Delta_1^2 \middle| w^{(\eta)}(t)\right] + \mathcal{O}(\eta^2) \\
\le&4\eta\frac{\left(w_i^{(\eta)}(t)\right)^{2\mu_1}}{\left(w_1^{(\eta)}(t)\right)^{2\mu_i}}\mathbb{E}\Big[\frac{\left((w^{(\eta)})^\top\tilde{\Lambda}^{(k)}w^{(\eta)}\right)^2\left(e_i^\top(\hat{\Lambda}_B^{(k)})\left(\Lambda^B\right)^{-\frac{1}{2}}w^{(\eta)}\right)^2 + \mu_i\left(e_i\tilde{\Lambda}^{(k)}w^{(\eta)}\right)^2}{(w_i^{(\eta)})^2} \\
&\cdot\mu_i\mu_1^2 + \frac{\left((w^{(\eta)})^\top\tilde{\Lambda}^{(k)}w^{(\eta)}\right)^2\left(e_1^\top(\hat{\Lambda}_B^{(k)})\left(\Lambda^B\right)^{-\frac{1}{2}}w^{(\eta)}\right)^2 + \mu_1\left(e_1\tilde{\Lambda}^{(k)}w^{(\eta)}\right)^2}{(w_1^{(\eta)})^2}\mu_1\mu_i^2\Big| w^{(\eta)}(t)\Big] + \mathcal{O}(\eta^2) \\
\le&4\eta^{2\delta}\left(C\frac{C_0C_1}{\mu_{\min}^2}D^3\mu_i\mu_1^2 + \mu_i^2\mu_1^2\frac{C_0}{\mu_{\min}}D\right) + \mathcal{O}(\eta) \\
=&\mathcal{O}(\eta^{2\delta}) \xrightarrow{\eta\to 0} 0,
\end{aligned}
$$

where the second inequality holds because of the mean inequality and the last inequality is from the independence of $A^{(k)}$ and $B^{(k)}$, $(w^{(\eta)})^\top\tilde{\Lambda}w^{(\eta)} \le \|\tilde{\Lambda}\|_2(w^{(\eta)})^\top w^{(\eta)} \le \frac{\|A^{(k)}\|_2}{\mu_{\min}}D$, since $\tilde{\Lambda}$ is symmetric, and $C = \frac{\left(w_i^{(\eta)}(t)\right)^{2\mu_1-2}}{\left(w_1^{(\eta)}(t)\right)^{2\mu_i}}$. By Section 4 of Chapter 7 in [17], we have that when $t \in [0,T]$, as $\eta \to 0$, $\frac{(w_i^{(\eta)})^{\mu_1}}{(w_1^{(\eta)})^{\mu_i}}$ weakly converges to the solution of (2.4.5) if they have the same initial solutions. Then, let $T \to \infty$, we know the convergence of $\frac{(w_i^{(\eta)})^{\mu_1}}{(w_1^{(\eta)})^{\mu_i}}$ holds at any time $t$. Note we can replace 1 by $j$, where $j \neq i$, and the proof still holds.

Moreover, using the same techniques, we can show that for all $i \in [d]$, $w_i^{(\eta)}$ converges to the solution of the following equation:

$$\frac{dw_i}{dt} = \mu_i(\beta_i - \sum_{j=1}^d \beta_j w_j^2)w_i. \tag{B.2.2}$$

Note that if any $w_i > 1$, $\mu_i(\beta_i - \sum_{j=1}^d \beta_j w_j^2)w_i < 0$, and if $\sum_{j=1}^d w_j^2 < 1$, $\mu_1(\beta_1 -$

$\sum_{j=1}^{d} \beta_j w_j^2)w_1 > 0$, which means that $w_1$ will increase. This further indicates that $w_1$ converges to $1$, while $w_i$ converges to 0 for all $i \neq 1$. This shows our algorithm converges to the neighbor of the global optima. □

### B.2.2 Proof of Lemma 2.4.3

*Proof.* We prove this by contradiction. Assume the conclusion does not hold, that is there exists a constant $C > 0$, such that for any $\eta' > 0$ we have

$$\sup_{\eta \leq \eta'} \mathbb{P}(\sup_t |z_i^{(\eta)}(t)| \leq C) = 1.$$

That implies there exists a sequence $\{\eta_n\}_{n=1}^{\infty}$ converging to $0$ such that

$$\lim_{n \to \infty} \mathbb{P}(\sup_t |z_i^{(\eta_n)}(t)| \leq C) = 1. \tag{B.2.3}$$

Thus, condition (i) in Theorem 2.4 [182] holds. We next check the second condition. When $\sup_t |z_i^{(\eta_n)}(t)| \leq C$ holds, Assumption 2.4.1 yields that $Z_i^{(\eta_n,k+1)} - z_i^{(\eta_n,k)} = C'\eta_n$, where $C'$ is some constant. Thus, for any $t, \epsilon > 0$, we have

$$|z_i^{(\eta_n)}(t) - Z_i^{(\eta_n)}(t+\epsilon)| = \frac{\epsilon}{\eta} C'\eta = C'\epsilon.$$

Thus, condition (ii) in Theorem 2.4 [182] holds. Then we have $\{Z_i^{(\eta_n)}(\cdot)\}_n$ is tight and thus converges weakly. We then calculate the infinitesimal conditional expectation

$$\begin{aligned}
&\frac{d}{dt}\mathbb{E}(z_j^{(\eta)}(t)) \\
&= \frac{1}{\eta}\mathbb{E}\left(z_j^{(\eta)}(t+\eta) - z_j^{(\eta)}(t) \middle| z_j^{(\eta)}(t)\right) = \eta^{-\frac{3}{2}}\mathbb{E}\left(w_j^{(\eta)}(t+\eta) - w_j^{(\eta)}(t) \middle| w_j^{(\eta)}(t)\right) \\
&= -\eta^{-\frac{1}{2}}\left[\left(w^{(\eta)}(t)\right)^\top \left(\Lambda^B\right)^{-\frac{1}{2}} (\Lambda^A) \left(\Lambda^B\right)^{-\frac{1}{2}} w^{(\eta)}(t) \cdot \left(\Lambda^B\right) w^{(\eta)}(t) - (\Lambda^A) w^{(\eta)}(t)\right]_j \\
&= \lambda_i z_j - \beta_i \mu_j z_j + \mathcal{O}(\eta^{1-2\delta}).
\end{aligned}$$

The last equality holds due to the fact that our initial point is near the saddle point $w_i^{(\eta)}(t) \approx e_i$ and $|w_j^{(\eta)}(t)| \leq C\eta^{\frac{1}{2}+\delta}$ . Next, we turn to the infinitesimal conditional variance,

$$
\begin{aligned}
&\tfrac{1}{\eta}\mathbb{E}\left[\left(z_j^{(\eta)}(t+\eta) - z_j^{(\eta)}(t)\right)^2 |z_j^{(\eta)}(t)\right] \\
=&\mathbb{E}\left[\left(e_j^\top \left(\left(\Lambda^B\right)^{\frac{1}{2}} \hat{\Lambda}_B^{(k)} \left(\Lambda^B\right)^{-\frac{1}{2}} w^{(k)} w^{(l)\top} - \Lambda^B\right) \cdot \tilde{\Lambda} w^{(k)}\right)^2 |w^{(k)}\right] \\
=&\mathbb{E}\left[\left(\left(\hat{\Lambda}_B^{(k)}\right)_{j,i} \cdot \sqrt{\mu_j/\mu_i} \cdot \tilde{\Lambda}_{i,i} - \mu_j \tilde{\Lambda}_{j,i}\right)^2\right] + \mathcal{O}(\eta^{3-6\delta}) \\
=&G_{j,i} + \mathcal{O}(\eta^{3-6\delta}) \leq 2\left(\tfrac{\mu_1}{\mu_j} \cdot C_0 \cdot C_1 + \mu_i^2 \cdot C_1\right).
\end{aligned}
$$

Then, we get the limit stochastic differential equation,

$$dz_j(t) = (-\beta_j \mu_i \cdot z_i + \lambda_i z_i)\, dt + \sqrt{G_{j,i}} dB(t) \ \text{ for } j \in [d] \backslash \{i\}.$$

Therefore, $\{Z_i^{(\eta_n)}(\cdot)\}$ converges weakly to a process defined by the equation above, which is an unstable O-U process with mean $0$ and exploding variance. Thus, for any $\tau$, there exist a time $t'$, such that

$$\mathbb{P}(|z_i(t')| \geq C) \geq 2\tau.$$

Since $\{z_i^{(\eta_n)}\}_n$ converges weakly to $z_i$, thus $\{z_i^{(\eta_n)}(t')\}_n$ converges in distribution to $Z_(t')$. This implies that there exists an $N > 0$, such that for any $n > N$

$$|\mathbb{P}(|z_i(T)| \geq C) - \mathbb{P}(|z_i^{(\eta_n)}(T)| \geq C)| \leq \tau.$$

Then we find a $t'$ such that

$$\mathbb{P}(|z_i^{(\eta_n)}(t')| \geq C) \geq \tau, \forall n > N,$$

or equivalently

$$\mathbb{P}(|z_i^{(\eta_n)}(t')| \leq C) < 1 - \tau, \forall n > N.$$

Since $\left\{\omega \middle| \sup_t |z_i^{(\eta_n)}(t)(\omega)| \leq C\right\} \subset \left\{\omega \middle| |z_i^{(\eta_n)}(\tau')(\omega) < C\right\}$, we have

$$\mathbb{P}(\sup_t |w_i^{(\eta_n)}(t)| \leq C\sqrt{\eta_n}) = \mathbb{P}(\sup_t |z_i^{(\eta_n)}(t)| \leq C) \leq 1 - \delta, \forall n > N,$$

which leads to a contradiction with (B.2.3). Our assumption does not hold.

□

### B.2.3 Proof of Lemma 2.4.4

*Proof.* Suppose the initial is near the stable equilibria, i.e., $|w_1^{(\eta)}(0) - 1| \leq C\eta^{\frac{1}{2}+\delta}$ and $|w_j^{(\eta)}(0)| \leq C\eta^{\frac{1}{2}+\delta}$ for all $j \neq 1$.First we show that $\|w^{(\eta)}(t)\|_2 \to 1$ as $t \to \infty$. With update (2.4.4), we show $w^{(\eta)\top}w^{(\eta)}(t)$ weakly converges to the following ODE by a similar proof in Lemma 2.4.2:

$$\begin{aligned}
&\frac{d}{dt}\mathbb{E}\left(w^{(\eta)\top}w^{(\eta)}(t)\right) \\
&= -w^{(\eta)\top}\left(\Lambda^B\right)^{-\frac{1}{2}}\left(\Lambda^A\right)\left(\Lambda^B\right)^{-\frac{1}{2}}w^{(\eta)} \cdot w^{(\eta)\top}\Lambda^B w^{(\eta)} + w^{(\eta)\top}(\Lambda^A)w^{(\eta)} + \mathcal{O}(\eta) \\
&= -\lambda_1\left(\|w^{(\eta)}\|_2^4 - \|w^{(\eta)}\|_2^2\right) + \mathcal{O}(\eta^{1-2\delta}),
\end{aligned}$$

Similarly, we can bound the infinitesimal conditional variance. Therefore, the norm of $w$ weakly converges to the following ODE:

$$dx = -\lambda_1\left(x^2 - x\right)dt.$$

The solution of the above ODE is

$$x = \begin{cases} \frac{1}{1-\exp(-\lambda_1 t + C)} & \text{if } x > 1 \\ \frac{1}{1+\exp(-\lambda_1 t + C)} & \text{if } x < 1 \\ 1 & \text{if } x = 1 \end{cases}.$$

This implies that $\|w^{(\eta)}(t)\|_2$ converges to 1 as $t \to \infty$. Then we calculate the infinitesimal conditional expectation for $i \neq 1$

$$
\begin{aligned}
&\frac{d}{dt}\mathbb{E}(z_i^{(\eta)}(t)) \\
&= \frac{1}{\eta}\mathbb{E}\left(z_i^{(\eta)}(t+\eta) - z_i^{(\eta)}(t) \middle| z_i^{(\eta)}(t)\right) = \eta^{-\frac{3}{2}}\mathbb{E}\left(w_i^{(\eta)}(t+\eta) - w_i^{(\eta)}(t) \middle| w_i^{(\eta)}(t)\right) \\
&= -\eta^{-\frac{1}{2}}\left[\left(w^{(\eta)}(t)\right)^\top \left(\Lambda^B\right)^{-\frac{1}{2}} \left(\Lambda^A\right) \left(\Lambda^B\right)^{-\frac{1}{2}} w^{(\eta)}(t) \cdot \left(\Lambda^B\right) w^{(\eta)}(t) - (\Lambda^A) w^{(\eta)}(t)\right]_i \\
&= \lambda_i z_i - \beta_1 \mu_i z_i + \mathcal{O}(\eta^{1-2\delta}).
\end{aligned}
$$

The last equality is from the fact that our initial point is near an optimum. Next, we turn to the infinitesimal conditional variance,

$$
\begin{aligned}
&\tfrac{1}{\eta}\mathbb{E}\left[\left(z_i^{(\eta)}(t+\eta) - z_i^{(\eta)}(t)\right)^2 \middle| z_i^{(\eta)}(t)\right] \\
&= \mathbb{E}\left[\left(e_i^\top \left(\left(\Lambda^B\right)^{\frac{1}{2}} \hat{\Lambda}_B^{(k)} \left(\Lambda^B\right)^{-\frac{1}{2}} w^{(k)} w^{(l)\top} - \Lambda^B\right) \cdot \tilde{\Lambda} w^{(k)}\right)^2 \middle| w^{(k)}\right] \\
&= \mathbb{E}\left[\left(\left(\hat{\Lambda}_B^{(k)}\right)_{i,1} \cdot \sqrt{\mu_i/\mu_1} \cdot \tilde{\Lambda}_{1,1} - \mu_i \tilde{\Lambda}_{i,1}\right)^2\right] + \mathcal{O}(\eta^{3-6\delta}) \\
&= G_{i,1} + \mathcal{O}(\eta^{3-6\delta}) \leq 2\left(\tfrac{\mu_i}{\mu_1} \cdot C_0 \cdot C_1 + \mu_i^2 \cdot C_1\right).
\end{aligned}
$$

By Section 4 of Chapter 7 in [17], we have that the algorithm converges to the solution of (2.4.10) if it is already near our optimal solution. □

### B.2.4 Proof of Theorem 2.4.5

*Proof.* Assume the initial is near a saddle point, $e_i$. According to Lemma 2.4.3 and (2.4.8), we obtain the closed form solution of (2.4.8) as follows:

$$
\begin{aligned}
z_j(t) &= z_j(0)\exp\left(-\mu_j\left(\beta_i - \beta_j\right)t\right) + \sqrt{G_{j,i}}\int_0^t \exp\left(\mu_j\left(\beta_i - \beta_j\right)(s-t)\right) dB(s) \\
&= \Bigg(\underbrace{z_j(0) + \sqrt{G_{j,i}}\int_0^t \exp\left(\mu_j(\beta_i - \beta_j)s\right) dB(s)}_{Q_1}\Bigg) \underbrace{\exp\left(-\mu_j\left(\beta_i - \beta_j\right)t\right)}_{Q_2}.
\end{aligned}
$$

We consider $j = 1$. Note at time $t$, $Q_1$ essentially is a random variable with mean $z_1(0)$ and variance $\frac{G_{1,i}\mu_1}{2(\beta_1-\beta_i)}\big(1 - \exp\big(-2\mu_1(\beta_1-\beta_i)t\big)\big)$, which has an upper bound $\frac{G_{1,i}\mu_1}{2(\beta_1-\beta_i)}$. $Q_2$, however, amplifies the magnitude of $Q_1$. Then it forces the algorithm escaping from the saddle point $e_i$. We consider the event $\{w_1(t)^2 > \eta\}$ and a random variable $v(t) \sim N\Big(0, \frac{G_{1,i}\mu_1}{2(\beta_1-\beta_i)}\big(\exp\big(2\mu_1(\beta_1-\beta_i)t\big) - 1\big)\Big)$. Because $z_j(0)$ might not be $0$, we have

$$\mathbb{P}(w_1(t)^2 > \eta) \geq \mathbb{P}(v^2(t) > 1).$$

Let the right hand side of (B.2.8) larger than $95\%$. Then with a sufficiently small $\eta$, we need

$$T_1 \asymp \frac{1}{\mu_1(\beta_1-\beta_i)}\log(\frac{200(\beta_1-\beta_i)}{\mu_1 G_{1,i}} + 1) \tag{B.2.4}$$

such that $\mathbb{P}(|w_1^{(\eta)}(T_1)|_2^2 > \eta) = 90\%$.

Now we consider the time required to converge under the ODE approximation.

By Lemma 2.4.2 with $j = 1$, after restarting the counter of time, we have

$$\frac{w_1^{\mu_i}(t)}{w_i^{\mu_1}(t)} \geq \eta^{\mu_i/2}\exp(\mu_1\mu_i(\beta_1-\beta_i)t).$$

Let the right hand side equal to $1$. Then with a sufficiently small $\eta$ we need

$$T_2 \asymp \frac{\mu_{\max}}{\mu_1\mu_{\min}\cdot \mathsf{gap}}\log(\eta^{-1}). \tag{B.2.5}$$

such that $\mathbb{P}\left(\frac{w_i^{(\eta)\mu_1}(T_2)}{w_1^{(\eta)\mu_i}(T_2)} \leq 1\right) = \frac{5}{6}$.

Then let $i = 1$ in Lemma 2.4.2. After restarting the counter of time, we have

$$\begin{aligned}\frac{w_i^{\mu_1}(t)}{w_1^{\mu_i}(t)} &\leq C\exp(\mu_{\max})\exp(\mu_1\mu_i(\beta_i-\beta_1)t)\\ \Longrightarrow w_i^2 &\leq \left(C\exp(\mu_{\max})\exp(\mu_1\mu_i(\beta_i-\beta_1)t)\right)^{2/\mu_1}\end{aligned}$$

where $\exp(\mu_{\max})$ comes from the above stage and $C$ is a constant containing $G_{1,i}$ and

$G_{i,j}$. The second inequality holds due to the fact that $w_1 \leq 1$, mentioned in the proof of Lemma 2.4.2. Therefore, given $\sum_{i=2}^d w_i^2 \leq \kappa\eta^{1+2\delta}$ and a sufficiently small $\eta$, we need

$$T_2' \asymp \frac{\mu_{\max}}{\mu_1\mu_{\min}\cdot \mathsf{gap}} \log(\eta^{-1}) \tag{B.2.6}$$

such that $\mathbb{P}\left(\frac{|w_1^{(\eta)}(T_2')|^2}{\|w^{(\eta)}(T_2')\|_2^2} > 1-\kappa\eta^{1+2\delta}\right) = \frac{8}{9}$.

Then the algorithm goes into Phase III. According to Lemma 2.4.4 and (2.4.10), we obtain the closed form solution of (2.4.10) as follows:

$$z_i(t) = z_i(0)\exp\left(-\mu_i\left(\beta_1-\beta_i\right)t\right) + \sqrt{G_{i,1}}\int_0^t \exp\left(\mu_i\left(\beta_1-\beta_i\right)(s-t)\right)dB(s).$$

By the Ito isometry property of the Ito-Integral, we have

$$\mathbb{E}\left(z_i(t)\right)^2 = \left(z_i(0)\right)^2 e^{-2\mu_i(\beta_1-\beta_i)t} + \frac{G_{i,1}}{2\mu_i\left(\beta_1-\beta_i\right)}\left[1-e^{-2\mu_i(\beta_1-\beta_i)t}\right]. \tag{B.2.7}$$

Then we consider the complement of the event $\{w_1^2 > 1-\epsilon\}$. By Markov inequality, we have

$$\begin{aligned}
&\mathbb{P}(w_1^2 \leq 1-\epsilon)\\
=&\mathbb{P}\left(\sum_{i=2}^d w_i^2 \geq \epsilon\right) \leq \frac{\mathbb{E}\left(\sum_{i=2}^d w_i^2\right)}{\epsilon} = \frac{\mathbb{E}\left(\sum_{i=2}^d z_i^2\right)}{\eta^{-1}\epsilon}\\
=&\frac{1}{\eta^{-1}\epsilon}\left(\sum_{i=2}^d \left(z_i(0)\right)^2 e^{-2\mu_i(\beta_1-\beta_i)t} + \frac{G_i}{2\mu_i\left(\beta_1-\beta_i\right)}\left[1-e^{-2\mu_i(\beta_1-\beta_i)t}\right]\right)\\
\leq&\frac{1}{\eta^{-1}\epsilon}\left(\eta^{-1}\delta^2 e^{-2\mu_{\min}\cdot\mathsf{gap}\cdot\mathsf{t}} + \frac{\phi}{2\mu_{\min}\cdot\mathsf{gap}}\right).
\end{aligned} \tag{B.2.8}$$

Let the right hand side of (B.2.8) be no larger than $\frac{1}{16}$.

$$\frac{1}{\eta^{-1}\epsilon}\left(\eta^{-1}\delta^2 e^{-2\mu_{\min}\cdot\mathsf{gap}\cdot t}+\frac{\phi}{2\mu_{\min}\cdot\mathsf{gap}}\right)\leq\frac{1}{16}$$
$$\implies e^{2\mu_{\min}\cdot\mathsf{gap}\cdot t}\geq\frac{16\cdot\mu_{\min}\cdot\mathsf{gap}\cdot\delta^2}{\epsilon\cdot\mu_{\min}\cdot\mathsf{gap}-16\cdot\eta\cdot\phi}.$$

Then after restarting the counter of time, we need

$$T_3\asymp\frac{1}{\mu_{\min}\cdot\mathsf{gap}}\cdot\log\left(\frac{\mu_{\min}\cdot\mathsf{gap}\cdot\delta^2}{\epsilon\cdot\mu_{\min}\cdot\mathsf{gap}-16\cdot\eta\cdot\phi}\right).\tag{B.2.9}$$

such that $\mathbb{P}(w_1^2(T_3)\geq 1-\epsilon)\geq\frac{15}{16}$.

Combining (B.2.4), (B.2.5), (B.2.6), (B.2.9), if our algorithm start from a saddle, then with probability at least $\frac{5}{8}$, we need

$$T=T_1+T_2+T_2'+T_3\asymp\frac{\mu_{\max}/\mu_{\min}}{\mu_1\cdot\mathsf{gap}}\log\left(\eta^{-1}\right)\tag{B.2.10}$$

such that $w_1^2(T)>1-\epsilon$.

Moreover, we choose

$$\eta\asymp\frac{\epsilon\cdot\mu_{\min}\cdot\mathsf{gap}}{\phi}.\tag{B.2.11}$$

Combining (B.2.10) and (B.2.11) together, we get the asymptotic sample complexity

$$N\asymp\frac{T}{\eta}\asymp\frac{\phi\cdot\mu_{\max}/\mu_{\min}}{\epsilon\cdot\mu_1\cdot\mu_{\min}\cdot\mathsf{gap}^2}\log\left(\frac{\phi}{\epsilon\cdot\mu_{\min}\cdot\mathsf{gap}}\right)\tag{B.2.12}$$

such that with probability at least $\frac{5}{8}$, we have $\|\hat{W}-W^*\|_2^2\leq\epsilon$. $\square$

# APPENDIX C

# SUPPLEMENTARY MATERIALS IN CHAPTER 3

## C.1 Detailed Proofs in Section 3.3

### C.1.1 Proof of Proposition 3.3.2

*Proof.* By Lemma 3.3.1, the posterior distribution follows a non-standardized t-distribution:

$$\Big[f(\mathbf{x})\big|\mathcal{D}_n\Big] \sim T\Big(2a+n-q, \hat{f}_n(\mathbf{x}), \tilde{\sigma}_n s_n(\mathbf{x})\Big).$$

Let $\nu_n = 2a+n-q$. The density function of $\Big[f(\mathbf{x})\big|\mathcal{D}_n\Big]$ then takes the following form:

$$g(f;\nu_n,\hat{f}_n,\tilde{\sigma}_n,s_n) = \frac{\Gamma((\nu_n+1)/2)}{\tilde{\sigma}_n s_n\sqrt{\nu_n\pi}\cdot\Gamma(\nu_n/2)}\left(1+\frac{(f-\hat{f}_n)^2}{\nu_n\tilde{\sigma}_n^2 s_n^2}\right)^{-(\nu_n+1)/2}.$$

Using this density function, the HEI criterion can then be simplified as:

$$\begin{aligned}\mathrm{HEI}_n(\mathbf{x}) &= \mathbb{E}_{f|\mathcal{D}_n}(y_n^*-f(\mathbf{x}))_+\\ &= \left(y_n^*-\hat{f}_n\right)\Phi_{\nu_n}\left(\frac{y_n^*-\hat{f}_n}{\tilde{\sigma}_n s_n}\right)+\int_{-\infty}^{y_n^*}\left(\hat{f}_n-f\right)g(f)df. \end{aligned} \tag{C.1.1}$$

The second term in (C.1.1) can be further simplified as:

$$
\begin{aligned}
&\int_{-\infty}^{y_n^*}\left(\hat{f}_n - f\right) g(f) df \\
=&\frac{\tilde{\sigma}_n s_n \nu_n}{\nu_n - 1}\frac{\Gamma((\nu_n+1)/2)}{\sqrt{\nu_n \pi}\cdot\Gamma(\nu_n/2)}\left(1+\frac{(y_n^*-\hat{f}_n)^2}{\nu_n(\tilde{\sigma}_n s_n)^2}\right)^{-\frac{\nu_n-1}{2}} \\
=&\frac{\sqrt{\nu_n}\tilde{\sigma}_n s_n \Gamma((\nu_n-1)/2)}{(\nu_n-2)\sqrt{\pi}\cdot\Gamma((\nu_n-2)/2)}\left(1+\frac{1}{\nu_n}\left(\frac{y_n^*-\hat{f}_n}{\tilde{\sigma}_n s_n}\right)^2\right)^{-\frac{\nu_n-1}{2}} \\
=&\sqrt{\frac{\nu_n}{\nu_n-2}}\tilde{\sigma}_n s_n \phi_{\nu_n-2}\left(\frac{y_n^*-\hat{f}_n}{\sqrt{\nu_n/(\nu_n-2)}\tilde{\sigma}_n s_n}\right).
\end{aligned}
\tag{C.1.2}
$$

Therefore, we prove the claim. □

### C.1.2 Proof of Proposition 3.3.3

If necessary, we denote the correlation function by $s_n(\mathbf{x};\boldsymbol{\theta})$ to highlight the dependence of $\boldsymbol{\theta}$. If not specified, we use simplified $s_n(\mathbf{x})$ to make general arguments.

The proof uses several Lemmas in [69]. The following lemma provides a lower bound for $s_n(\mathbf{x};\boldsymbol{\theta})$.

**Lemma C.1.1** (Lemma 7 in [69])**.** Set $\zeta = \alpha$ if $\nu \leq 1$ otherwise 0. Given $\boldsymbol{\theta} \in \mathbb{R}_+^d$, there is a constant $C' > 0$ depending only on $\Omega, K$ and $\boldsymbol{\theta}$ which satisfies the following:
For any $k \in \mathbb{N}$, and sequences $\mathbf{x}_n \in \Omega$, $\tilde{\boldsymbol{\theta}}_n \geq \boldsymbol{\theta}$, the inequality

$$s_n(\mathbf{x}_{n+1};\tilde{\boldsymbol{\theta}}_n) \geq C' k^{-(\nu\wedge 1)/d}(logk)^{\zeta}$$

holds for at most $k$ distinct $n$.

**Lemma C.1.2** (Lemma 9 in [69])**.** Given $\boldsymbol{\theta}^L, \boldsymbol{\theta}^U \in \mathbb{R}_+^d$, pick sequences $\mathbf{x}_n \in \Omega$, the corresponding posterior of scale parameter $\boldsymbol{\theta}^L \leq \tilde{\boldsymbol{\theta}}_n \leq \boldsymbol{\theta}^U$. Then for an open $S \subset \Omega$,

$$\sup_{x\in S} s_n(x;\tilde{\boldsymbol{\theta}}_n) = \boldsymbol{\Omega}(n^{-\nu/d}), \tag{C.1.3}$$

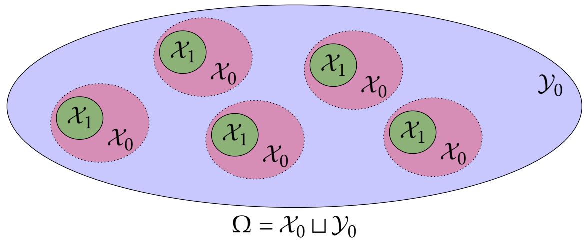



Figure C.1: An illustrative example of how domain $\Omega$ is partitioned to two disjoint parts: an open set $\mathcal{X}_0$ and a closed set $\mathcal{Y}_0$. $\mathcal{X}_1$ is a closed subset of $\mathcal{X}_0$ (each green circle is a subset of $\mathcal{X}_1$ and each pink circle is a subset of $\mathcal{X}_0$). Under the distribution $F$ over $\Omega$, the probability of choosing $\mathcal{X}_0$ is less than $\epsilon$.

uniformly in the sequences $x_n, \tilde{\boldsymbol{\theta}}_n$. Here $\boldsymbol{\Omega}$ denotes the asymptotic lower bound notation.

*Proof.* This is a constructive proof. The key idea is that given the initial points $\mathbf{x}_1, \cdots, \mathbf{x}_k$ independent of the objective function, we construct two function $h(\mathbf{x})$ and $\tilde{h}(\mathbf{x})$ such that SEI cannot distinguish these two functions and misses the global optimal of $\tilde{h}$.

First, given a random initial strategy $F$, we can use a union of small open subsets, denoted by $\mathcal{X}_0$, such that $k$ initial points generated by the strategy satisfies that $P_F(\mathbf{x}_1, ..., \mathbf{x}_k \in \mathcal{X}_0) \leq \epsilon$. Then we partition the domain $\Omega$ as shown in Figure C.1. Here $\mathcal{X}_0$ and $\mathcal{Y}_0$ are two disjoint non-empty interior domains. Then we first construct a function $h$ as follows:

$$h(\mathbf{x}) := \begin{cases} 0 & \mathbf{x} \in \mathcal{Y}_0 \\ 1 & \mathbf{x} \in \mathcal{X}_1 \\ h_1(\mathbf{x}) & \mathbf{x} \in \mathcal{X}_0 \backslash \mathcal{X}_1 \end{cases}. \tag{C.1.4}$$

Here $h_1(\mathbf{x}) \geq 0$ is a function that ensures $h(\mathbf{x}) \in C^\infty(\Omega)$. Thus, it is easy to verify that $h(\mathbf{x}) \in \mathcal{H}_{\boldsymbol{\theta}}(\Omega)$. We denote $\|h\|_{\mathcal{H}_{\boldsymbol{\theta}}(\Omega)} = R^2$. With such a function $h(\mathbf{x})$, if there is one $i \in \{1, 2, ..., k\}$ such that $\mathbf{x}_i \in \mathcal{Y}_0$, then $y_k^* = 0$. Therefore, the probability of $y_k^* = 0$ is at least $1 - \epsilon$. Then we show that conditioning on $y_k^* = 0$, denoted by event $A_0$, SEI cannot visit $\mathcal{X}_1$ infinitely often. We prove this by contradiction. By Lemma C.1.1, we know that as $n \to \infty$, $s_n(\mathbf{x}_{n+1}; \tilde{\boldsymbol{\theta}}_n) \to 0$. Suppose $\mathbf{x}_{n+1} \in \mathcal{X}_1$. If $s_n(\mathbf{x}_{n+1}; \tilde{\boldsymbol{\theta}}_n) = 0$, then $\mathbf{x}_{n+1} \in \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ and $\text{SEI}_n(\mathbf{x}_{n+1}) = 0$. Therefore, we can find a $\mathbf{z}_{n+1} \in \mathcal{Y}_0$ such that

$s_n(\mathbf{z}_{n+1}) \neq 0$ and $\mathrm{SEI}_n(\mathbf{x}_{n+1}) = 0 < \mathrm{SEI}_n(\mathbf{z}_{n+1})$. If $s_n(\mathbf{x}_{n+1}; \tilde{\boldsymbol{\theta}}_n) > 0$, then we have

$$T_n(\mathbf{x}_{n+1}) := \frac{y_n^* - h(\mathbf{x}_{n+1})}{s_n(\mathbf{x}_{n+1}; \tilde{\boldsymbol{\theta}}_n)} = -s_n(\mathbf{x}_{n+1}; \tilde{\boldsymbol{\theta}}_n)^{-1} \to -\infty.$$

SEI [73] has the following form:

$$\begin{aligned} \mathrm{SEI}_n(\mathbf{x}) =& \tilde{\sigma}_n s_n(\mathbf{x}) \Big( \frac{I_n(\mathbf{x})}{\tilde{\sigma}_n s_n(\mathbf{x})} \Phi_n \big( \frac{I_n(\mathbf{x})}{\tilde{\sigma}_n s_n(\mathbf{x})} \big) \\ &+ \frac{\nu_n + (I_n(\mathbf{x})/\tilde{\sigma}_n s_n(\mathbf{x}))^2}{\nu_n - 1} \phi_n \big( \frac{I_n(\mathbf{x})}{\tilde{\sigma}_n s_n(\mathbf{x})} \big) \Big). \end{aligned} \tag{C.1.5}$$

Moreover, $T_n$ and $I_n$ are connected by Lemma C.3.3, i.e.,

$$I_n(\mathbf{x})/s_n(\mathbf{x}) \leq T_n(\mathbf{x}) + \|h\|_{\mathcal{H}_{\tilde{\boldsymbol{\theta}}_n}(\Omega)} \leq T_n(\mathbf{x}) + R\sqrt{\prod_{i=1}^{d} \theta_i^U / \theta_i^L} =: T_n(\mathbf{x}) + S.$$

The last inequality holds because of Lemma C.3.1. Then we establish an upper bound for the part in parentheses in (C.1.5):

$$\begin{aligned} x\Phi_n(x) + \frac{\nu_n + x^2}{\nu_n - 1}\phi_n(x) &= \Theta \left( -\nu_n^{-(\nu_n+1)/2} \frac{x^{-\nu_n - 1}}{\nu_n + 2} + 2\nu_n^{-(\nu_n+1)/2} \frac{|x|^{-\nu_n+1}}{\nu_n} \right) \\ &= \Theta(|x|^{-\nu_n+1}) \quad \text{as } x \to -\infty. \end{aligned} \tag{C.1.6}$$

The first equality holds because of the fact $(1 + x^2/\nu_n) \geq \frac{x^2}{\nu_n}$. On the other hand, by Lemma C.1.2, there exists a $\mathbf{z}_{n+1} \in \mathcal{Y}_0$ such that $h(\mathbf{z}_{n+1}) = 0$ and $s_n(\mathbf{z}_{n+1}; \tilde{\boldsymbol{\theta}}_n) = \boldsymbol{\Omega}(n^{-\nu/d})$. Then we have

$$\mathrm{SEI}_n(\mathbf{z}_{n+1}; \tilde{\boldsymbol{\theta}}_n) = \tilde{\sigma}_n s_n(\mathbf{z}_{n+1}; \tilde{\boldsymbol{\theta}}_n) \left( \frac{\nu_n}{\nu_n - 1} \frac{\Gamma((\nu_n+1)/2)}{\sqrt{\nu_n \pi}\Gamma(\nu_n/2)} \right).$$

Here $\Gamma$ denotes the gamma function. Therefore, we obtain

$$\frac{\mathrm{SEI}_n(\mathbf{x}_{n+1}; \tilde{\boldsymbol{\theta}}_n)}{\mathrm{SEI}_n(\mathbf{z}_{n+1}; \tilde{\boldsymbol{\theta}}_n)} \leq \mathcal{O} \left( \frac{((T_n(\mathbf{x}) + S)/\tilde{\sigma}_n)^{-\nu_n+1}}{n^{-\nu/d}} \right) = o(1). \tag{C.1.7}$$

The last equality holds because $T_n \to -\infty$ and $\nu_n$ is the same order as $n$ under a fixed hyperparameter. Therefore, by contradiction, we know that on $A_0$, there is a random variable $w$, for all $n > w$, $x_n \notin \mathcal{X}_1$. Hence there is a constant $t \in \mathbb{N}$, depending on $\epsilon$, such that the event $A_1 = A_0 \cap \{w \leq t\}$ has probability at least $1 - 2\epsilon$ under the SEI strategy. Then we can further select an open set $\mathcal{W} \subset \mathcal{X}_1$ such that the event $A_2 = A_1 \cap \{\mathbf{x}_n \notin \mathcal{W} : n < t\}$ has probability at least $1 - 3\epsilon$.

Finally, we can construct another smooth function $g(\mathbf{x})$ like $h(\mathbf{x})$, which equals $0$ when $\mathbf{x} \notin \mathcal{W}$ and has minimum $-2$. Then we construct $\tilde{h}(\mathbf{x}) := h(\mathbf{x}) + g(\mathbf{x})$, which has minimum $-1$. However, SEI cannot distinguish the difference between these two functions and $\inf_n \tilde{h}(\mathbf{x}_n) \geq 0$. Thus, for $\delta = 1$, we have

$$\mathbb{P}_F\left(\inf_n \tilde{h}(\mathbf{x}_n^*) - \min_{\mathbf{x}\in\Omega} \tilde{h}(\mathbf{x}) \geq \delta\right) \geq 1 - 3\epsilon.$$

We obtain the desired result. $\square$

## C.2 Detailed Proofs in Section 3.4

### C.2.1 Proof of Proposition 3.4.1

*Proof.* The marginal likelihood can be obtained by integrating out the parameters $\boldsymbol{\beta}$ and $\sigma^2$ in the hierarchical model :

$$\begin{aligned}
&p(\mathbf{y}_n; a, b)\\
=&\int \frac{\exp\{-(\mathbf{y}_n - \mathbf{P}_n\boldsymbol{\beta})^\top \mathbf{K}_n^{-1}(\mathbf{y}_n - \mathbf{P}_n\boldsymbol{\beta})/(2\sigma^2)\}}{\sqrt{2\pi \det(\sigma^2\mathbf{K}_n)}} \frac{(b/\sigma^2)^a}{\sigma^2\Gamma(a)} \exp\left(-\frac{b}{\sigma^2}\right) d\boldsymbol{\beta} d\sigma^2\\
=&\int \sqrt{\frac{\det\left(\sigma^2\mathbf{G}_n^{-1}\right)}{\det\left(\sigma^2\mathbf{K}_n\right)}} \exp\left\{-\frac{\mathbf{y}_n^\top\mathbf{K}_n^{-1}\mathbf{y}_n - \hat{\boldsymbol{\beta}}_n^\top\mathbf{G}_n\hat{\boldsymbol{\beta}}_n}{2\sigma^2}\right\} \frac{(b/\sigma^2)^a}{\sigma^2\Gamma(a)} \exp\left(-\frac{b}{\sigma^2}\right) d\sigma^2\\
=&\sqrt{\frac{\det(\mathbf{G}_n^{-1})}{\det(\mathbf{K}_n)}} \frac{b^a}{\Gamma(a)} \int (\sigma^2)^{-(a+\frac{n-q}{2})-1} \exp\left\{-\frac{\mathbf{y}_n^\top\mathbf{K}_n^{-1}\mathbf{y}_n - \hat{\boldsymbol{\beta}}_n^\top\mathbf{G}_n\hat{\boldsymbol{\beta}}_n + 2b}{2\sigma^2}\right\} d\sigma^2\\
=&\sqrt{\frac{\det(\mathbf{G}_n^{-1})}{\det(\mathbf{K}_n)}} \frac{b^a}{\Gamma(a)} \frac{\Gamma(a + (n-q)/2)}{(b + (\mathbf{y}_n^\top\mathbf{K}_n^{-1}\mathbf{y}_n - \hat{\boldsymbol{\beta}}_n^\top\mathbf{G}_n\hat{\boldsymbol{\beta}}_n)/2)^{a+\frac{n-q}{2}}}.
\end{aligned}$$

Consider next the optimization of the marginal likelihood (3.4.2). Since the first term $\sqrt{\det(\mathbf{G}_n^{-1})/\det(\mathbf{K}_n)}$ does not involve $a$ and $b$, we consider only the remaining terms in (3.4.2), and denote it as $p(\mathbf{y}_n; a, b)$. The partial derivative of $p(\mathbf{y}_n; a, b)$ in $b$ is:

$$\nabla_b\, p(\mathbf{y}_n; a, b) = \frac{\Gamma(a+(n-q)/2)}{\Gamma(a)} \frac{b^{a-1}(a(\mathbf{y}_n^\top \mathbf{K}_n^{-1}\mathbf{y}_n - \hat{\boldsymbol{\beta}}_n^\top \mathbf{G}_n \hat{\boldsymbol{\beta}}_n)/2 - b(n-q)/2)}{(b+(\mathbf{y}_n^\top \mathbf{K}_n^{-1}\mathbf{y}_n - \hat{\boldsymbol{\beta}}_n^\top \mathbf{G}_n \hat{\boldsymbol{\beta}}_n)/2)^{a+(n-q)/2+1}}. \tag{C.2.1}$$

Setting this to zero and solving for $b$, we get the profile maximizer:

$$b^*(a) = a\left(\mathbf{y}_n^\top \mathbf{K}_n^{-1}\mathbf{y}_n - \hat{\boldsymbol{\beta}}_n^\top \mathbf{G}_n \hat{\boldsymbol{\beta}}_n\right)/(n-q). \tag{C.2.2}$$

Now, let $w = \mathbf{y}_n^\top \mathbf{K}_n^{-1}\mathbf{y}_n - \hat{\boldsymbol{\beta}}_n^\top \mathbf{G}_n \hat{\boldsymbol{\beta}}_n$, in which case $b^*(a) = a \cdot w/(n-q)$. With this, the (rescaled) marginal likelihood can be written as a function of only $a$:

$$p(\mathbf{y}_n; a, b^*(a)) = \frac{(aw)^a \Gamma(a+(n-q)/2)}{\Gamma(a)(n-q)^a (aw/(n-q)+w/2)^{a+\frac{n-q}{2}}}.$$

Taking the gradient of $p(\mathbf{y}_n; a, b^*(a))$ in $a$, we get:

$$\begin{aligned}\nabla_a\, p(\mathbf{y}_n; a, b^*(a)) &= \frac{(aw)^a \Gamma(a+\frac{n-q}{2})}{\Gamma(a)(aw/(n-q)+w/2)^{a+(n-q)/2}(n-q)^a} \\ &\quad \cdot \left(\Psi\left(a+\frac{n-q}{2}\right) - \Psi(a) - \log\left(1+\frac{n-q}{2a}\right)\right),\end{aligned} \tag{C.2.3}$$

where $\Psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ satisfies $\Psi(a+1) = \frac{1}{a} + \Psi(a)$. Therefore, for even values of $n-q$, we have

$$\Psi\left(a+\frac{n-q}{2}\right) - \Psi(a) = \sum_{i=0}^{(n-q)/2-1} \frac{1}{a+i} \geq \log\left(1+\frac{n-q}{2a}\right),$$

while for odd values of $n-q$, we have

$$\begin{aligned}\Psi\left(a+\frac{n-q}{2}\right)-\Psi(a) &\geq \sum_{i=0}^{(n-q-1)/2-1}\frac{1}{a+i}+\frac{1}{2a+n-q-1}\\ &\geq \log\left(1+\frac{n-q-1}{2a}\right)+\frac{1}{2a+n-q-1}\\ &\geq \log\left(1+\frac{n-q}{2a}\right).\end{aligned} \tag{C.2.4}$$

Hence, $p(\mathbf{y}_n; a, b^*(a))$ is a monotonically increasing function in $a$, and it follows that there are no finite maximizer for the marginal likelihood $p(\mathbf{y}_n; a, b)$ over $(a, b) \in \mathbb{R}_+^2$.

### C.2.2 Proof of Proposition 3.4.2

With the hyperpriors $[a] \sim \Gamma(\zeta, \iota)$ and $[b] \propto \mathbf{1}$, the profile maximizer (C.2.2) still holds. MMAP then aims to maximize

$$\tilde{p}(\mathbf{y}_n; a, b^*(a)) = \frac{(aw)^a\Gamma(a+\frac{n-q}{2})(n-q)^{-a}}{\Gamma(a)(aw/(n-q)+w/2)^{a+\frac{n-q}{2}}}\frac{\iota^\zeta a^{\zeta-1}\exp(-\iota\cdot a)}{\Gamma(c)}. \tag{C.2.5}$$

Calculating the derivative of logarithm (C.2.5), we obtain

$$\nabla_a \log\tilde{p}(\mathbf{y}_n; a, b^*(a)) = \psi(a+\tfrac{n-q}{2}) - \psi(a) - \log(1+\tfrac{n-q}{2a}) + \tfrac{\zeta-1}{a} - \iota,$$

which is a decreasing function with $\lim_{a\to\infty}\nabla_a \log\tilde{p}(\mathbf{y}_n; a, b^*(a)) < 0$. This guarantees a finite solution for the MMAP optimization problem over $(a, b) \in \mathbb{R}_+^2$. □

## C.3 Detailed Proofs in Section 3.5

### C.3.1 Proof of Theorem 3.5.3

The proof of Theorem 3.5.3 requires the following three lemmas. The first lemma provides an upper bound for the RKHS norm of function $f$ for changing scale parameters:

**Lemma C.3.1** (Lemma 4 in [69])**.** If $f \in \mathcal{H}_{\boldsymbol{\theta}}(\Omega)$, then $f \in \mathcal{H}_{\boldsymbol{\theta}'}(\Omega)$ for all $\mathbf{0} < \boldsymbol{\theta}' \leq \boldsymbol{\theta}$, and

$$\|f\|^2_{\mathcal{H}_{\boldsymbol{\theta}'}(\Omega)} \leq \left(\prod_{i=1}^d \theta_i/\theta_i'\right) \|f\|^2_{\mathcal{H}_{\boldsymbol{\theta}}(\Omega)}.$$

The following two lemmas describe the posterior distribution of $f$ with trend in terms of $\mathcal{H}_{\boldsymbol{\theta}}(\Omega)$. For simplicity, we denote $k_{\mathbf{x}_i} = K_{\boldsymbol{\theta}}(\mathbf{x}_i, \cdot) \in \mathcal{H}_{\boldsymbol{\theta}}(\Omega)$ for $i = 1, ..., n$.

**Lemma C.3.2.** Suppose $f \in \mathcal{H}_{\boldsymbol{\theta}}(\Omega)$, and $\mathbf{p}(\mathbf{x}) \in \mathcal{H}_{\boldsymbol{\theta}}(\Omega)$. Let $g(\mathbf{x}) = f(\mathbf{x}) - \mathbf{p}(\mathbf{x})^\top \boldsymbol{\beta}$. Then the estimates $\hat{f}_n(\mathbf{x})$ and $\hat{\boldsymbol{\beta}}_n$ in Lemma 3.3.1 solves the following optimization problem:

$$\min_{\boldsymbol{\beta}, g(\mathbf{x})} \|g\|^2_{\mathcal{H}_{\boldsymbol{\theta}}(\Omega)} \tag{C.3.1}$$

$$\text{subject to} \quad \mathbf{p}(\mathbf{x}_i)^\top \boldsymbol{\beta} + g(\mathbf{x}_i) = f(\mathbf{x}_i), \quad i = 1, \cdots, n,$$

with minimum value $n\hat{\sigma}_n^2$.

*Proof.* Since $\Omega$ is a compact domain, $g(\mathbf{x}) = f(\mathbf{x}) - \mathbf{p}(\mathbf{x})^\top \boldsymbol{\beta}$, still belongs to the space $\mathcal{H}_{\boldsymbol{\theta}}(\Omega)$. Let $W = \text{Span}(k_{\mathbf{x}_1}, \ldots, k_{\mathbf{x}_n})$, and decompose $g = g^\parallel + g^\perp$, where $g^\parallel \in W$, $g^\perp \in W^\perp$, the orthogonal complement space of $W$. It follows that $g^\perp(\mathbf{x}_i) = \langle g^\perp, k_{\mathbf{x}_i} \rangle = 0$, Since $g^\perp$ affects the optimization only through $\|g\|_{\mathcal{H}_{\boldsymbol{\theta}}(\Omega)}$, the minimizer must satisfy $g^\perp = 0$.

We can now represent $g$ as $g = \sum_{i=1}^n \upsilon_i k_{\mathbf{x}_i}$, for some $\upsilon_i \in \mathbb{R}$, $i = 1, ..., n$. The optimization problem (C.3.1) then becomes

$$\min_{\boldsymbol{\upsilon}, \boldsymbol{\beta}} \boldsymbol{\upsilon}^\top \mathbf{K}_n \boldsymbol{\upsilon} \quad \text{subject to} \quad \mathbf{P}_n \boldsymbol{\beta} + \mathbf{K}_n \boldsymbol{\upsilon} = \mathbf{y}_n,$$

which gives the estimates in Theorem 3.3.1. □

The third lemma gives a useful upper bound on the difference between the true function $f$ and the GP predictor $\hat{f}_n$:

**Lemma C.3.3** (Theorem 11.4 in [87])**.** For $f \in \mathcal{H}_{\boldsymbol{\theta}}(\Omega)$, the GP predictor $\hat{f}_n$ has the following pointwise error bound:

$$|f(\mathbf{x}) - \hat{f}_n(\mathbf{x})| \leq s_n(\mathbf{x})\|f\|_{\mathcal{H}_{\boldsymbol{\theta}}(\Omega)}. \tag{C.3.2}$$

With these lemmas in hand, we now proceed with the proof of Theorem 3.5.3:

*Proof.* Recall, we denote $I_n(\mathbf{x}) = y_n^* - \hat{f}_n(\mathbf{x})$. For simplicity, we further denote $u_n(\mathbf{x}) = (y_n^* - f(\mathbf{x}))_+$ and

$$\tau_n(x) = x\Phi_{\nu_n}(x) + \sqrt{\frac{\nu_n}{\nu_n - 2}} \cdot \phi_{\nu_n - 2}\left(\frac{x}{\sqrt{\nu_n/(\nu_n - 2)}}\right).$$

The HEI criterion can then be written as:

$$\mathrm{HEI}_n(\mathbf{x}) = \tilde{\sigma}_n s_n(\mathbf{x})\tau_n\left(\tfrac{I_n(\mathbf{x})}{\tilde{\sigma}_n s_n(\mathbf{x})}\right). \tag{C.3.3}$$

Since $\tau_n'(x) = \Phi_{\nu_n}(x) \geq 0$, $\tau_n(x)$ must be non-decreasing in $x$. Moreover, we denote $R = \|f\|_{\mathcal{H}_{\boldsymbol{\theta}^U}(\Omega)}$, then by Lemma C.3.1, we have $\|f\|_{\mathcal{H}_{\tilde{\boldsymbol{\theta}}_n}(\Omega)} \leq R\sqrt{\prod_{i=1}^d \frac{\theta_i^U}{\theta_i^L}} := S$, and by Lemma C.3.3, if $u_n(\mathbf{x}) > 0$, then $|u_n(\mathbf{x}) - I_n(\mathbf{x})| \leq s_n(\mathbf{x})S$. Thus,

$$\mathrm{HEI}_n(\mathbf{x}) \geq \tilde{\sigma}_n s_n(\mathbf{x})\tau_n\left(\tfrac{u_n(\mathbf{x}) - Ss_n(\mathbf{x})}{\tilde{\sigma}_n s_n(\mathbf{x})}\right) \geq \tilde{\sigma}_n s_n(\mathbf{x})\tau_n\left(\tfrac{-S}{\tilde{\sigma}_n}\right). \tag{C.3.4}$$

Note that $\tau_n(x) = x - x\Phi_{\nu_n}(-x) + \sqrt{\nu_n/(\nu_n - 2)}\phi_{\nu_n - 2}(x/\sqrt{\nu_n/(\nu_n - 2)}) = x + \tau_n(-x)$. Therefore,

$$\mathrm{HEI}_n(\mathbf{x}) \geq \tilde{\sigma}_n s_n(\mathbf{x})\tau_n\left(\tfrac{u_n(\mathbf{x}) - Ss_n(\mathbf{x})}{\tilde{\sigma}_n s_n(\mathbf{x})}\right) \geq u_n(\mathbf{x}) - Ss_n(\mathbf{x}). \tag{C.3.5}$$

On the one hand, by inequalities (C.3.4) and (C.3.5), we have the following lower bound on $\mathrm{HEI}_n(\mathbf{x})$:

$$\mathrm{HEI}_n(\mathbf{x}) \geq \tfrac{\tilde{\sigma}_n \tau_n(-S/\tilde{\sigma}_n)}{S + \tilde{\sigma}_n \tau_n(-S/\tilde{\sigma}_n)} u_n(\mathbf{x}) \geq \tfrac{\tau_n(-S/\tilde{\sigma}_n)}{\tau_n(S/\tilde{\sigma}_n)} u_n(\mathbf{x}). \tag{C.3.6}$$

Note the above inequality also holds for $u_n(\mathbf{x}) = 0$. Therefore it holds for any $u_n(\mathbf{x})$.

On the other hand, note that $\tau_n(x) = x + \tau_n(-x) \leq x + \tau_n(0)$ for any $x \geq 0$. Moreover $\tau_n(0) = \sqrt{\nu_n/(\nu_n - 2)}\phi_{\nu_n - 2}(0) \leq 2$, since $n > q + 1$, $\nu_n/(\nu_n - 2) \leq 3$ and $\phi_{\nu_n}(0) \leq \phi(0) \leq 2/5$. Thus, $\tau(x) \leq x + 2$ for $x \geq 0$. Plugging this into (C.3.3), we get the following upper bound on $\mathrm{HEI}_n(\mathbf{x})$:

$$\mathrm{HEI}_n(\mathbf{x}) \leq \tilde{\sigma}_n s_n(\mathbf{x}) \tau_n\left(\frac{u_n(\mathbf{x}) + S s_n(\mathbf{x})}{\tilde{\sigma}_n s_n(\mathbf{x})}\right) \leq u_n(\mathbf{x}) + (S + 2\tilde{\sigma}_n)\, s_n(\mathbf{x}). \tag{C.3.7}$$

By Lemma 7 of Bull (2011), we know that there exists a constant $C_2$, depending on $\Omega$, $K$, and $\boldsymbol{\theta}^L$ such that for any sequence $\mathbf{x}_n \in \Omega$ and $k \in \mathbb{N}$, the inequality

$$s_n(\mathbf{x}_{n+1}) \geq C_2 k^{-(\nu \wedge 1)/d} (\log k)^{\zeta}$$

holds at most $k$ times.

Furthermore, since $\|f\|_{\mathcal{H}_{\boldsymbol{\theta}^U}(\Omega)} = R$, we have

$$\textstyle\sum_{i=1}^{n} (y_i^* - y_{i+1}^*) = y_1^* - y_{n+1}^* \leq y_1^* - \min_{\mathbf{x} \in \Omega} f(\mathbf{x}) \leq 2\|f\|_\infty \leq 2R.$$

Therefore, by $y_n^* - y_{n+1}^* \geq 0$, it follows that $y_n^* - y_{n+1}^* \geq 2Rk^{-1}$ holds at most $k$ times. Otherwise, the above inequality does not hold. Furthermore, by $y_{n+1}^* \leq f(\mathbf{x}_{n+1})$, we have $y_n^* - f(\mathbf{x}_{n+1}) \geq 2Rk^{-1}$ holds at most $k$ times. Thus, there exists an $n_k \in \mathbb{N}$, with $k \leq n_k \leq 3k$, for which

$$s_{n_k}(\mathbf{x}_{n_k+1}) \leq C_2 k^{-(\nu \wedge 1)/d} (\log k)^{\zeta} \quad \text{and} \quad u_{n_k}(\mathbf{x}_{n_k+1}) \leq 2Rk^{-1}.$$

Since $y_n^*$ is non-increasing in $n$, for $3k \leq n < 3(k+1)$, we further have

$$\begin{aligned}
y_n^* - f(\mathbf{x}^*) &\leq y_{n_k}^* - f(\mathbf{x}^*) \\
&\leq \frac{\tau_{n_k}(S/\tilde{\sigma}_{n_k})}{\tau_{n_k}(-S/\tilde{\sigma}_{n_k})} \mathrm{HEI}_{n_k}(\mathbf{x}^*) \\
&\leq \frac{\tau_{n_k}(S\tilde{\sigma}_{n_k})}{\tau_{n_k}(-S\tilde{\sigma}_{n_k})} \mathrm{HEI}_{n_k}(\mathbf{x}_{n_k+1}) \\
&\leq \frac{\tau_{n_k}(S/\tilde{\sigma}_{n_k})}{\tau_{n_k}(-S/\tilde{\sigma}_{n_k})} \left(2Rk^{-1} + C_2\left(S + 2\tilde{\sigma}_{n_k}\right) k^{-(\nu\wedge 1)/d} (\log k)^{\zeta}\right) \\
&\leq \frac{\tau_{n_k}(S/c)}{\tau_{n_k}(-S/c)} \left(2Rk^{-1} + C_3 S k^{-(\nu\wedge 1)/d} (\log k)^{\zeta}\right), \qquad \text{(C.3.8)}
\end{aligned}$$

where the second last inequality holds from Lemma C.3.2 and the last inequality holds from Lemma C.3.1 since $\tilde{\sigma}_{n_k}$ is based on the MAP estimate $\tilde{\boldsymbol{\theta}}_n$. From this, we obtain the desired result

$$f(\mathbf{x}_n^*) - \min_{\mathbf{x}} f(\mathbf{x}) = \begin{cases} \mathcal{O}(n^{-\nu/d}(\log n)^{\alpha}), & \nu \leq 1, \\ \mathcal{O}(n^{-1/d}), & \nu > 1. \end{cases} \qquad \text{(C.3.9)}$$

□

### C.3.2 Proof of Theorem 3.5.5

With Assumption 3.5.4, the picked points are quasi-uniform distributed by Theorems 14 and 18 in [183]. Therefore, with large enough $n$, the fill distance can be small enough. Then by Theorem 1 in [90], we obtain the desired results.

# APPENDIX D
# SUPPLEMENTARY MATERIALS IN CHAPTER 4

## D.1 Limiting Cycle

Limiting cycle is a well-known issue for bilevel machine learning problems [4,5]. The reason behind limiting cycle is that different from minimization problems, a minimax optimization problem is more complicated and can be highly nonconvex-nonconcave, where the inner problem can not be solved exactly. Here we provide a concrete example that even for a convex-concave problem, the inexact solutions can result in a limiting cycle: We consider the following optimization problem:

$$\min_x \max_y f(x, y) = xy.$$

Then at the $t$-th iteration, the update direction will be $(-y_t, x_t)$. If we start from $(1, 0)$ with a stepsize of $0.0001$, this update will result in a limiting circle: $x^2 + y^2 = 1$ and never reach the stable equilibrium $(0, 0)$ as shown in Figure D.1.

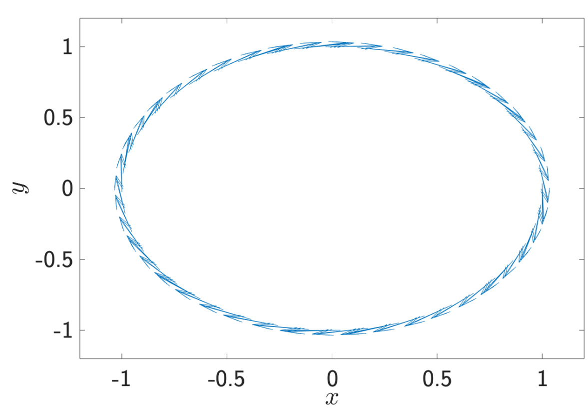



Figure D.1: An example of the limiting circle: arrows denote the update directions

### D.2 Black-box Attack

Under the black-box setting, we first train a surrogate model with the same architecture of the target model but a different random seed, and then attackers generate adversarial examples to attack the target model by querying gradients from the surrogate model.

The black-box attack highly relies on the transferability, which is the property that the adversarial examples of one model are likely to fool others. However, the transferred attack is very unstable, and often has a large variation in its effectiveness. Therefore, results of the black-box setting might not be reliable and effective. Thus we only present one result here to demonstrate the robustness of different models.

Table D.1: Results of the black-box setting over CIFAR-10. We evaluate L2L methods with slim attacker networks.

| Surrogate | Plain Net | | FGSM Net | | PGM Net | |
|---|---|---|---|---|---|---|
| | FGSM | PGM10 | FGSM | PGM10 | FGSM | PGM10 |
| Plain Net | 40.03 | 5.60 | 74.42 | 75.25 | 67.37 | 65.92 |
| FGSM Net | 79.20 | 85.02 | **89.90** | 80.40 | 64.28 | 63.89 |
| PGM Net | 83.80 | 84.73 | 84.33 | 85.29 | 67.05 | 65.54 |
| Naive L2L | 45.52 | 25.95 | 83.99 | 77.94 | 68.14 | 67.13 |
| Grad L2L | **86.10** | 86.87 | 87.93 | **88.01** | **71.15** | **69.95** |
| 2-Step L2L | 85.83 | **87.10** | 86.51 | 87.60 | 70.58 | 69.38 |

Table D.2: Experiments under the black-box setting over CIFAR-100. Note that here we only evaluate L2L methods using the slim attacker network.

| Surrogate | Plain Net | | FGSM Net | | PGM Net | |
|---|---|---|---|---|---|---|
| | FGSM | PGM10 | FGSM | PGM10 | FGSM | PGM10 |
| Plain Net | 21.04 | 9.04 | 50.57 | 54.06 | 40.06 | 41.30 |
| FGSM Net | 42.87 | 50.73 | **61.68** | 44.70 | 39.34 | 40.08 |
| PGM Net | 56.63 | 58.34 | 56.99 | 57.97 | 40.19 | 39.87 |
| Naive L2L | 20.97 | 10.47 | 50.36 | 54.07 | 38.63 | 39.91 |
| Grad L2L | 57.63 | 59.62 | 59.18 | **61.26** | 41.71 | 41.15 |
| 2-Step L2L | **58.66** | **59.31** | 58.92 | 59.46 | **45.80** | **45.31** |

### D.3 Slim Network

Table D.3 presents another architecture that we used in the L2L. In this network, the second convolutional layer uses downsampling, while the second last deconvolutional layer uses upsampling. Due to the downsampling, this network is computationally cheap and thus it is computationally fast. For example the running time of per epoch for L2L with slim attacker is 480; whereas L2L with the original architecture is 620. However, it loses some information of input and is less stable than the original architecture (Table 4.1). Inspired by residual learning in [102], we address the above issues by using a skip layer connection to ease the training of this network. Specifically, the last layer takes the concatenation of $\mathcal{A}_{f_\theta}(\boldsymbol{x}, y)$ and the output of the second last layer as input. Figure D.2 presents the architecture of ResBlocks. PReLU is a special type of Leaky ReLU with a learnable slope parameter.

Table D.3: Slim Attacker Network Architecture.

| | |
|---|---|
| Conv: | $[k = 3 \times 3, c = 128, s = 1, p = 1]$, BN+ReLU |
| ResBlocks: | [channel = 256] |
| ResBlocks: | [channel = 128], BN |
| DeConv: | $[k = 4 \times 4, c = 16, s = 2, p = 1]$, BN+ReLU |
| Conv: | $[k = 3 \times 3, c = 3, s = 1, p = 1]$, tanh |

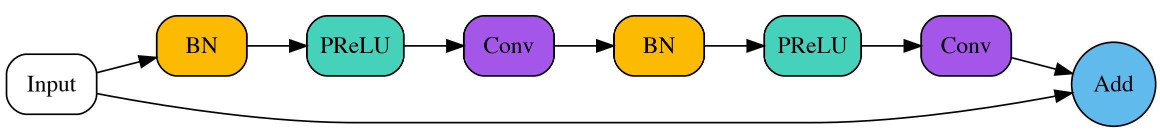



Figure D.2: An illustration example for the architecture of ResBlocks.

Table D.4 shows the results of L2L with the architecture shown in Table D.3.

Table D.4: Results of L2L with slim attacker under white-box setting over CIFAR.

| | **CIFAR-10** | | | | **CIFAR-100** | | | |
|---|---|---|---|---|---|---|---|---|
| | Clean | FGSM | PGM20 | CW | Clean | FGSM | PGM20 | CW |
| Naive L2L | 94.41 | 28.44 | 0.01 | 0.00 | 75.27 | 8.47 | 0.05 | 0.00 |
| Grad L2L | 85.31 | 57.44 | 53.02 | 42.72 | 60.60 | 26.58 | 27.37 | 23.14 |
| 2-Step L2L | 75.36 | 60.19 | 46.12 | 40.82 | 60.23 | 25.92 | 20.23 | 22.70 |

## D.4 Robustness Evaluation Checklist

Recently, there are many works on robustness defense that have been proven ineffective [184, 185]. Our work follows the most reliable and widely used robust model approach adversarial training, which finds a set parameters to make the model robust. We do not make any modification to final classifier model. Unlike previous works (e.g., Defense-GAN, [186]), our model does not take the attacker as a part of the final model and does not use shattered/obfuscated/masked gradient as a defense mechanism. We also demonstrate that the evaluation of the robustness of our proposed L2L method is trustworthy by verifying all items listed in [185].

### D.4.1 Shattered/Obfuscated/Masked Gradient

In this section we verify that our proposed L2L method does not fall into the pitfall of shattered/obfuscated/masked gradient, which have proven ineffective. To see this, we checked every item recommended in section 3.1 of [184]:

- One-step attacks perform better than iterative attacks: Figure 4.4 shows that the PGM attack is stronger with larger number of iterations.
- Black-box attacks are better than white-box attacks: Appendix D.2 shows that the black-box transfer attack is much weaker than white white-box attacks.
- Unbounded attacks do not reach $100\%$ success: We evaluate the model robustness against attack with extremely large perturbation to show that unbounded attacks do reach $100\%$ success. Specifically, we use the PGM-10 attack with various perturbation magnitudes

$\epsilon \in [0, 1]$ and stepsize $\frac{\epsilon}{10}$. Figure D.3 shows that the PGM attack eventually reach $100\%$ success as the perturbation magnitude increases.

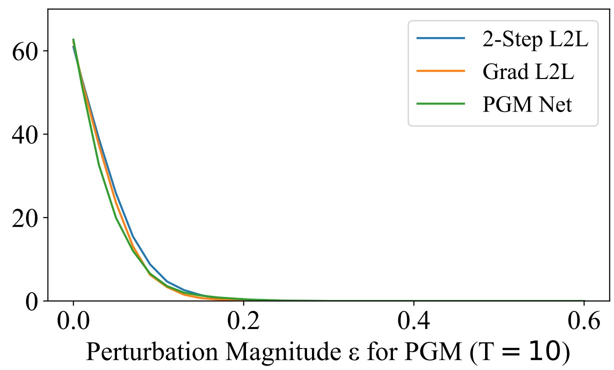



Figure D.3: Robust accuracy against perturbation magnitudes of PGM over CIFAR-100.

- Random sampling finds adversarial examples: In Table 4.2, we show that random search is not better than gradient-based method and is rather weak against our model.
- Increasing distortion bound does not increase success: Figure 4.4 shows that the PGM attack becomes stronger as the perturbation magnitude increases.

### D.4.2 Robustness Evaluation Checklist

[185] also provide an evaluation checklist, and we now check each of common severe flaws and common pitfalls as follows:

- State a precise threat model: We do not have any adversary detector; We do not use shattered/Obfuscated/Masked gradient. We do not have a denoiser. Our model has no aware of the attack mechanism, including PGM and CW attacks.
- Adaptive attacks: We used CW, PGM, and L2L attacker attack.
- Report clean model accuracy: We reported.
- Do not use Fast Gradient Sign Method. We use PGM-20 and PGM-100 and CW.

- Do not only use attacks during testing that were used during training. We use different evaluation criteria to evaluate all models.
- Perform basic sanity tests: It is provided in Figure 4.4.
- Generate an attack success rate vs. perturbation budget: Figure 4.4.
- Verify adaptive attacks perform better than any other (e.g., blackbox, and brute-force search): The above table and Appendix D.2 in the paper.
- Describe the attacks applied: In Section 4.4.
- Apply a diverse set of attacks: We tried PGM attack (with different perturbation magnitude and iterations), blackbox attack (transfer attack), CW attack (adaptive attack), L2L attack (adaptive and designed for this particular model), Bruteforce random search (gradient-free attack)
- Suggestions for randomized defenses: We are not.
- Suggestions for non-differentiable components (e.g., by performing quantization or adding extra randomness): We have no additional non-differentiable component.
- Verify that the attacks have converged: Figure 4.4 shows that the PGM attack eventually converges.
- Carefully investigate attack hyperparameters: Figure 4.4.
- Compare against prior work: We compared our algorithm to PDM net. L2L is more computationally efficient and the L2L model is more robust due to the fact that L2L attack is strong enough. Unlike Defense-GAN, we do not use the generator (attacker in L2L) as the denoising module and do not change the final prediction model.

### D.5 Extension

Our proposed L2L framework is quite general, and applicable to a *broad* class of minimax optimization problems. We present an extension of our proposed L2L framework to generative adversarial imitation learning (GAIL, [187]) and provide numerical experiments for comparing the original GAIL and GAIL with L2L on two environments: CartPole and Mountain Car [188].

#### D.5.1 L2L for Generative Adversarial Imitation Learning

GAIL aims to learn a policy from expert's behavior, by recovering the expert's cost function and extracting a policy from the recovered cost function, which can also be formulated as a bilevel optimization problem:

$$\begin{aligned} &\min_{\theta_\pi} \mathop{\mathbb{E}}_{s,a\sim\pi(s;\theta_\pi)} [\log (D(s,a;\theta_{\mathrm{D}}^*))] + \mathop{\mathbb{E}}_{\tilde{s},\tilde{a}\sim\pi_{\mathrm{E}}} [\log(1 - D(\tilde{s},\tilde{a};\theta_{\mathrm{D}}^*))] - \lambda H(\pi), \\ \text{s.t.} \quad &\theta_{\mathrm{D}}^* \in \mathop{\mathrm{argmax}}_{\theta_{\mathrm{D}}} \mathop{\mathbb{E}}_{s,a\sim\pi(s;\theta_\pi)} [\log (D(s,a;\theta_{\mathrm{D}}))] + \mathop{\mathbb{E}}_{\tilde{s},\tilde{a}\sim\pi_{\mathrm{E}}} [\log(1 - D(\tilde{s},\tilde{a};\theta_{\mathrm{D}}))] - \lambda H(\pi), \end{aligned} \tag{D.5.1}$$

where $\pi(\cdot;\theta_\pi)$ is the trained policy parameterized by $\theta_\pi$, $\pi_{\mathrm{E}}$ denotes the expert policy, $D(\cdot,\cdot;\theta_{\mathrm{D}})$ is the discriminator parameterized by $\theta_{\mathrm{D}}$, $\lambda H(\pi)$ denotes a entropy regularizer with tuning parameter $\lambda$, $(s,a)$ and $(\tilde{s},\tilde{a})$ denote the state-action for the trained policy and expert policy, respectively. By optimizing (D.5.1), the discriminator $D$ distinguishes the state-action $(s,a)$ generated from the learned policy $\pi$ with the sampled trajectories $(\tilde{s},\tilde{a})$ generated from some expert policy $\pi_{\mathrm{E}}$. In the original GAIL training, for each iteration, we update the parameter of $D$, $\theta_{\mathrm{D}}$, by stochastic gradient ascend and then update $\theta_\pi$ by the trust region policy optimization (TRPO, [189]).

Similar to the adversarial training with L2L, we apply our L2L framework to GAIL by parameterizing the inner optimizer as a neural network $U(;\theta_{\mathrm{U}})$ with parameter $\theta_{\mathrm{U}}$. Its input contains two parts: parameter $\theta_{\mathrm{D}}$ and the gradient of loss function with respect to $\theta_{\mathrm{D}}$:

$$g_{\mathrm{D}}(\theta_{\mathrm{D}}, \theta_{\pi}) = \mathop{\mathbb{E}}_{s,a\sim\pi(s;\theta_{\pi})}[\nabla_{\theta_{\mathrm{D}}} \log\left(D(s,a;\theta_{\mathrm{D}})\right)] + \mathop{\mathbb{E}}_{\tilde{s},\tilde{a}\sim\pi_E}[\nabla_{\theta_{\mathrm{D}}} \log(1 - D(\tilde{s},\tilde{a};\theta_{\mathrm{D}}))].$$

In practice, we use a minibatch (several sample trajectories) to estimate $g_{\mathrm{D}}(\theta_{\mathrm{D}}, \theta_{\pi})$, denoted as $\hat{g}_{\mathrm{D}}(\theta_{\mathrm{D}}, \theta_{\pi})$. Specifically, at the $t$-th iteration, we first calculate $\hat{g}^t_{\mathrm{D}} = \hat{g}_{\mathrm{D}}(\theta^t_{\mathrm{D}}, \theta^t_{\pi})$ and then update $\theta^{t+1}_{\mathrm{D}} = U(\theta^t_{\mathrm{D}}, \hat{g}^t_{\mathrm{D}}; \theta^t_{\mathrm{U}})$. Next, we update $\theta_{\mathrm{U}}$ by gradient ascend based on the sample estimate of

$$\mathop{\mathbb{E}}_{s,a\sim\pi(s;\theta^t_{\pi})}[\nabla_{\theta_{\mathrm{U}}} \log\left(D(s,a;\theta^{t+1}_{\mathrm{D}})\right)] + \mathop{\mathbb{E}}_{\tilde{s},\tilde{a}\sim\pi_{\mathrm{E}}}[\nabla_{\theta_{\mathrm{U}}} \log(1 - D(\tilde{s},\tilde{a};\theta^{t+1}_{\mathrm{D}}))].$$

The detailed algorithm is presented in Algorithm 8.

**Algorithm 8** Learning-to-learn-based generative adversarial imitation learning

**Input:** $\pi_E(\tilde{s})$: Expert; $\theta_\pi$: Policy; $\theta_{\mathrm{D}}$: Discriminator; $\theta_{\mathrm{U}}$: Updater.

**for** $t \leftarrow 1$ *to* $N$ **do**

Sample trajectories $(s, a \sim \pi(a;\theta_\pi))$ and expert trajectories $(\tilde{s}, \tilde{a} \sim \pi_E(\tilde{s}))$.

$g^t_{\mathrm{D}} \leftarrow \frac{1}{|(s,a)|}\sum_{(s,a)}[\nabla_{\theta_{\mathrm{D}}} \log\left(D(s,a;\theta^t_{\mathrm{D}})\right)] + \frac{1}{|(\tilde{s},\tilde{a})|}\sum_{(\tilde{s},\tilde{a})}[\nabla_{\theta_{\mathrm{D}}} \log(1 - D(\tilde{s},\tilde{a};\theta^t_{\mathrm{D}}))]$

//Compute gradient.

$\theta^{t+1}_{\mathrm{D}} = U(\theta^t_{\mathrm{D}}, g^t_{\mathrm{D}}; \theta^t_{\mathrm{U}})$

//Update the discriminator parameters.

$\theta^{t+1}_{\mathrm{U}} \leftarrow \mathop{\mathrm{argmin}}_{\theta_{\mathrm{U}}} \frac{1}{|(s,a)|}\sum_{(s,a)}[\log\left(D(s,a;\theta^{t+1}_{\mathrm{D}})\right)] + \frac{1}{|(\tilde{s},\tilde{a})|}\sum_{(\tilde{s},\tilde{a})}[\log(1 - D(\tilde{s},\tilde{a};\theta^{t+1}_{\mathrm{D}}))]$

//Update $\theta_{\mathrm{U}}$ of updater.

Update policy parameter $\theta_\pi$ by a policy step using the TRPO rule [187].

#### D.5.2 Numerical Experiments

**Updater Architecture.** We use a simple 3-layer perceptron with a skip layer as our updater. The number hidden units are ($2m \to 8m \to 4m \to m$), where $m$ is the dimension of $\theta_{\mathrm{D}}$ that depends on the original task. For the first and second layers, we use Parametric ReLU

(PReLU [190]) as the activation function, while the last layer has no activation function. Finally we add the output to $\theta_{\mathrm{D}}$ in the original input as the updated parameter for the discriminator network.

**Hyperparameter Settings.** For all baselines we exactly follows the setting in [187], except that we use a 2-layer discriminator with number of hidden units ($(s, a) \rightarrow 64 \rightarrow 32 \rightarrow 1$) using $\tanh$ as the activation function. We use the same neural network architecture for $\pi$ and the same optimizer configuration. The expert trajectories are obtained by an expert trained using TRPO. For L2L based GAIL, we also use Adam optimizer to update the $\theta_{\mathrm{U}}$ with the same configuration as updating $\theta_{\mathrm{D}}$ in the original GAIL.

**Numerical Results.** As can be seen in Figure D.4, GAIL has a sudden performance drop after training for a long time. We conjecture that this is because the discriminator overfits the expert trajectories and converges to a bad optimum, which is not generalizable. On the other hand, GAIL with L2L is much more stable. It is very important to real applications of GAIL: since the reward in real-world environment is usually unaccessible, we cannot know whether there is a sudden performance drop or not. With L2L, we can stabilize the training and obtain a much more reliable algorithm for real-world applications.

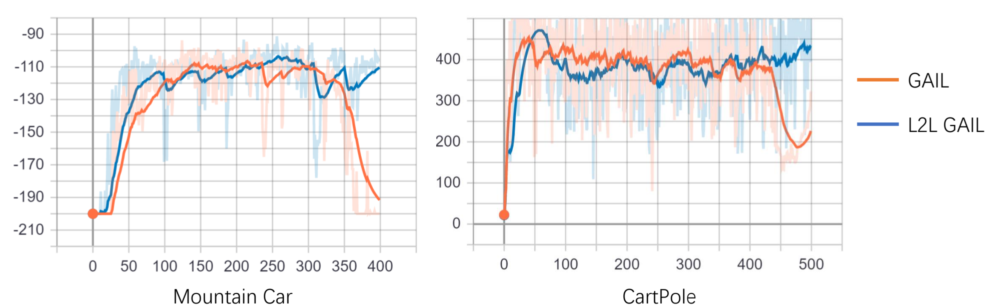



Figure D.4: Reward vs. iteration of the trained policy using original GAIL and L2L GAIL under two environments: Mountain Car and CartPole.

# APPENDIX E

# SUPPLEMENTARY MATERIALS IN CHAPTER 5

## E.1 Proof of Proposition 5.2.1

*Proof.* 1. We introduce the statistics

$$R_i^t = \sum_{k=1}^{t} \exp\left(\sum_{l=k}^{t} \frac{b}{\Sigma_{ii}}\left(\epsilon_i^l - \frac{b}{2}\right)\right), \quad \text{for } i = 1, ..., p,$$

and the auxiliary stopping time

$$\eta(s) = \inf\left\{t : \sum_{i=1}^{p} R_i^t \geq \exp(s)\right\}.$$

Then we have

$$\exp\left(\max_{i\in\{1,...,p\}} C_i^t\right) \leq \max_{i\in\{1,...,p\}} R_i^t \leq \sum_{i=1}^{p} R_i^t. \tag{E.1.1}$$

The first inequality holds since $R_i^t$ considers all the cumulative statistics. (E.1.1) then implies

$$\mathbb{E}_\infty T(s) \geq \mathbb{E}_\infty \eta(s) \geq \frac{\exp(s)}{p}.$$

The second inequality comes from Lemma 1 in [191].

2. We prove this by showing the right hand side is both the asymptotic lower and upper bounds of the EDD of $T$. Recall that

$$T_i(s) := \inf\left\{t : C_i^t \geq s\right\}, \quad \text{and} \quad T(s) = \min_{j=1}^{p}\{T_j(s)\}.$$

First, under alternative hypothesis $H_i$, we have that as $s \to \infty$,

$$\frac{\mathbb{E}_i(T(s))}{\frac{2\Sigma_{ii}s}{b^2}} \leq \frac{\mathbb{E}_i(T_i(s))}{\frac{2\Sigma_{ii}s}{b^2}} \sim 1.$$

The first inequality holds since by definition we have $T(s) \leq T_i(s)$ and the second asymptotic result is standard (See Section 6.5.1 in [192]). By taking the limit superior on both sides, we have

$$\limsup_{s\to\infty} \frac{\mathbb{E}_i(T(s))}{\frac{2\Sigma_{ii}s}{b^2}} \leq \limsup_{s\to\infty} \frac{\mathbb{E}_i(T_i(s))}{\frac{2\Sigma_{ii}s}{b^2}} = 1.$$

Therefore, we obtain it is the asymptotic upper bound. Then we prove that it is also the asymptotic lower bound. For notational simplicity, we denote that $T_i'(s) = \min_{j\neq i} T_j(s)$ as the minimum stopping time of nodes except node $v_i$. Then the stopping time $T = \min(T_i, T_i')$. Note that under the alternative hypothesis $H_A$ with $\mathcal{S} = \{i\}$, we have that $T_i'$ has the same distribution as that under the null hypothesis $H_0$. Therefore, by definition and total expectation, we have

$$\begin{aligned}
\mathbb{E}_i(T) &= \mathbb{E}_i(\min(T_i, T_i')) \\
&= \sup_{\tau\geq 1} \operatorname{ess\,sup}[\mathbb{E}_i^\tau(T_i - \tau)^+ \mathbb{P}_{i,\tau}(T_i \leq T_i') + \mathbb{E}_i^\tau(T_i' - \tau)^+ \mathbb{P}_{i,\tau}(T_i > T_i')],
\end{aligned}$$

where $\mathbb{P}_{i,\tau}$ denotes the probability under alternative hypothesis $H_A$ with change time $\tau$. Now let us consider $\mathbb{P}_{i,\tau}(T_i \leq T_i')$:

$$\begin{aligned}
\mathbb{P}_{i,\tau}\left(T_i \leq T_i'\right) &= \mathbb{P}_{i,\tau}\left(C_i^{T_i} \geq s \geq \max_{j\neq i, t\leq T_i} C_j^t\right) \\
&\geq \mathbb{P}_{i,\tau}\left(C_i^{T_i} \geq s\right) - \mathbb{P}_{i,\tau}(s \leq \max_{j\neq i, t\leq T_i} C_j^t) \\
&\geq 1 - (p-1)\exp(-s).
\end{aligned}$$

The last inequality holds because the Appendix 2 on Page 245 of [193] and part 1 of Proposition 5.2.1. Consequently, we prove that $\frac{2\Sigma_{ii}}{b^2} s$ is also the asymptotic lower bound.

Thus, we obtain the desired results.

$\square$

## E.2 Proof of Lemma 5.3.1

*Proof.* For a variance CAR model with mean $\boldsymbol{\mu}$, we have

$$\epsilon_i^t \big| \epsilon_j^t, j \neq i \sim \mathcal{N}\left(\frac{\phi}{W_{i+}} \sum_{j \in N(i)} \frac{\sigma_i}{\sigma_j}(\epsilon_j^t - \mu_j), \frac{\sigma_i^2}{W_{i+}}\right).$$

where $\mu_i$ is the $i$-th element of $\boldsymbol{\mu}$, e.g., under $H_\infty$, $\boldsymbol{\mu} = \mathbf{0}$ and under $H_A$ with $\mathcal{S} = \{i\}$, $\boldsymbol{\mu} = b \cdot \boldsymbol{e}_i$. Then by Brook's Lemma [160] and some algebraic manipulations, we have

$$\begin{aligned}
\frac{P_i(\boldsymbol{\epsilon}^t)}{P_\infty(\boldsymbol{\epsilon}^t)} &= \prod_{l=1}^{d} \frac{\frac{P_i(\epsilon_l^t|\epsilon_{j0}^t, \epsilon_k^t, 1 \leq j < l < k \leq d)}{P_i(\epsilon_{l0}^t|\epsilon_{j0}^t, \epsilon_k^t, 1 \leq j < l < k \leq d)}}{\frac{P_\infty(\epsilon_l^t|\epsilon_{j0}^t, \epsilon_k^t, 1 \leq j < l < k \leq d)}{P_\infty(\epsilon_{l0}^t|\epsilon_{j0}^t, \epsilon_k^t, 1 \leq j < l < k \leq d)}} \cdot \frac{P_i(\epsilon_{10}^t, \cdots, \epsilon_{d0}^t)}{P_\infty(\epsilon_{10}^t, \cdots, \epsilon_{d0}^t)} \\
&= \prod_{l \in N(i) \cup \{i\}} \frac{\frac{P_i(\epsilon_l^t|\epsilon_{j0}^t, \epsilon_k^t, 1 \leq j < l < k \leq d)}{P_i(\epsilon_{l0}^t|\epsilon_{j0}^t, \epsilon_k^t, 1 \leq j < l < k \leq d)}}{\frac{P_\infty(\epsilon_l^t|\epsilon_{j0}^t, \epsilon_k^t, 1 \leq j < l < k \leq d)}{P_\infty(\epsilon_{l0}^t|\epsilon_{j0}^t, \epsilon_k^t, 1 \leq j < l < k \leq d)}} \cdot \frac{P_i(\epsilon_{10}^t, \cdots, \epsilon_{d0}^t)}{P_\infty(\epsilon_{10}^t, \cdots, \epsilon_{d0}^t)} \\
&= \prod_{l \in N(i)} \exp\left(-\phi \frac{b(\epsilon_l^t - \epsilon_{l0}^t)}{\sigma_l^2}\right) \cdot \exp\left(\frac{W_{i+} b(\epsilon_i^t - \epsilon_{i0}^t)}{\sigma_i^2}\right) \\
&\quad \cdot \exp\left(b\mathbf{e}_i' \boldsymbol{\Lambda}^{-1/2} (\boldsymbol{D_W} - \phi \boldsymbol{W}) \boldsymbol{\Lambda}^{-1/2} (2\boldsymbol{\epsilon}_0^t - b\mathbf{e}_i)\right) \\
&= \exp\left(\frac{bW_{i+}}{\sigma_i^2}\Big(\epsilon_i^t - \frac{\phi}{W_{i+}} \sum_{j \in N(i)} \frac{\sigma_i}{\sigma_j}\epsilon_j - \frac{b}{2}\Big)\right).
\end{aligned} \tag{E.2.1}$$

The second equality in (E.2.1) holds since only the node $v_i$'s and its neighbors' conditional distribution changes, *i.e.*, for $j \notin N(i) \cup \{i\}$ $P_i(\epsilon_j|\epsilon_k, k \neq j)$ does not change. We obtain the result. $\square$

## E.3 Proof of Theorem 5.3.2

*Proof.* The proof of the first part is similar to that of Proposition 5.2.1. Here we focus on the second part. We first show the asymptotic upper bound. Note that

$$\mathbb{E}_i[\tilde{T}] \leq \mathbb{E}_i[\tilde{T}_i] \sim \frac{b\sigma_i^2}{c_i W_{i+}}.$$

The last step is the direct result of Lemma 5.3.1. With a similar argument to that of Proposition 5.2.1, we obtain the asymptotic upper bound. Now we show it is also the asymptotic lower bound. Denote $\mathbb{E}_i$ as the expectation under the alternative hypothesis $H_A$ with $\mathcal{S} = \{i\}$. We then compute the KL-divergence between $P_i$ and $P_\infty$:

$$\begin{aligned}\text{KL}(P_i, P_\infty) &= \mathbb{E}_i \log \frac{P_i(\boldsymbol{\epsilon}^t)}{P_\infty(\boldsymbol{\epsilon}^t)} \\ &= \mathbb{E}_i \left( \frac{c_i}{\sigma^2} \left( W_{i+}\epsilon_i^t - \phi \sum_{j \in N(i)} \epsilon_j^t - W_{i+}\frac{c_i}{2} \right) \right) \\ &= \frac{c_i^2 W_{i+}}{2\sigma^2}. \end{aligned} \tag{E.3.1}$$

By the Strong Law of Large Numbers, we have

$$\lim_{M\to\infty} \sup_{1\leq\tau<\infty} \text{ess sup}\, P_i\left( \frac{1}{M} \max_{1\leq l\leq M} \sum_{k=\tau+1}^{\tau+l} \log \frac{P_i(\boldsymbol{\epsilon}^t)}{P_\infty(\boldsymbol{\epsilon}^t)} \geq (1+\gamma)\text{KL}(P_i, P_\infty) \middle| \boldsymbol{\epsilon}^t, t = 1, ..., \tau \right) = 0, \tag{E.3.2}$$

holds for any $\gamma > 0$. To obtain our desired results, we use the Markov inequality [194]:

$$\mathbb{E}_{i,\tau}(\tilde{T} - \tau)^+ \geq \beta \frac{s}{\text{KL}(P_i, P_\infty)} P_{i,\tau}\left( \tilde{T} - \tau \geq \beta \frac{s}{\text{KL}(P_i, P_\infty)} \right) \tag{E.3.3}$$

holds for any $\beta > 0$. We then show that $P_{i,\tau}\left(\tilde{T} - \tau \geq \beta \frac{s}{\mathrm{KL}(P_i, P_\infty)}\right) \to 1$ for any $1 > \beta > 0$ and $\tau \geq 1$.

We denote $Z_i(k,l) = \sum_{t=l}^{k} \frac{P_i(\boldsymbol{\epsilon}^t)}{P_\infty(\boldsymbol{\epsilon}^t)}$. By change of measure we have:

$$
\begin{aligned}
J_i :&= P_\infty(0 \leq \tilde{T} - \tau \leq l) \\
&= \mathbb{E}_{i,\tau}\left[\mathbf{1}_{\{0 \leq \tilde{T} - \tau \leq l\}} \exp\left(-Z_i(\tilde{T}, \tau)\right)\right] \\
&\geq \mathbb{E}_{i,\tau}\left[\mathbf{1}_{\{0 \leq \tilde{T} - \tau \leq l, Z_i(\tilde{T}, \tau) < B\}} \exp\left(-Z_i(\tilde{T}, \tau)\right)\right] \\
&\geq \exp(-B) P_{i,\tau}\left(0 \leq \tilde{T} - \tau \leq l, \max_{\tau \leq n < \tau + l} Z_i(n, \tau) < B\right) \\
&\geq \exp(-B)\left[P_{i,\tau}\left(0 \leq \tilde{T} - \tau \leq l\right) - P_{i,\tau}\left(\max_{\tau \leq n < \tau + l} Z_i(n, \tau) \geq B\right)\right].
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
P_{i,\tau}(0 \leq \tilde{T} - \tau < l) \leq \exp(B) P_\infty\left(0 \leq \tilde{T} - \tau \leq l\right) \\
+ P_{i,\tau}\left(\max_{\tau \leq n < \tau + l} Z_i(n, \tau) \geq B\right).
\end{aligned} \tag{E.3.4}
$$

Let $l = \lfloor (1-\beta) \frac{s}{J_i} \rfloor$ and $B = (1-\beta^2)s$, then we have

$$
\begin{aligned}
P_{i,\tau}(\tau \leq \tilde{T} < \tau + l) \leq \exp\left((1-\beta^2)s\right) P_\infty\left(\tau \leq \tilde{T} < \tau + l\right) \\
+ \sup_{\tau \geq 1} P_{i,\tau}\left(\max_{\tau \leq n < \tau + \lfloor (1-\beta)\frac{s}{J_i} \rfloor} Z_i(n, \tau) \geq (1-\beta^2)s\right).
\end{aligned} \tag{E.3.5}
$$

By (E.3.2), we have

$$
\sup_{\tau \geq 1} P_{i,\tau}\left(\max_{\tau \leq n < \tau + \lfloor (1-\beta)\frac{s}{J_i} \rfloor} Z_i(n, \tau) \geq (1-\beta^2)s\right) \to 0. \tag{E.3.6}
$$

On the other hand, we introduce a virtual statistics:

$$
C'_t := \max\left(\max_{i \in [p]} \epsilon_i^t + C'_{t-1}, 0\right),
$$

and the corresponding stopping $T' := \inf_t\{C'_t \geq s\}$. Then we have: as $s \to \infty$

$$P_\infty(\tilde{T} \geq k) \geq P_\infty(T' \geq k) \geq 1 - k\exp(-s).$$

Therefore, we have

$$P_\infty(\tilde{T} < k) \leq k\exp(-s).$$

Then we have that

$$\begin{aligned}\exp((1-\beta^2)s)P_\infty(\tau \leq \tilde{T} < \tau + l) &\leq \exp((1-\beta^2)s)(\tau + l)\exp(-s)\\ &\leq (\tau + l)\exp(-\beta^2 s) \to 0.\end{aligned} \tag{E.3.7}$$

Let $\beta \to 1$. Then we obtain the desired result. □

## E.4 Proof of Proposition 5.3.3

*Proof.* First, under the CAR model (5.3.4), we have

$$\begin{aligned}\Sigma_{ii}/\sigma_i^2 = (\boldsymbol{D_W} - \phi\boldsymbol{W})_{ii}^{-1} &= \left(\boldsymbol{D_W}^{-1/2}(\boldsymbol{I} - \phi\boldsymbol{D_W}^{-1/2}\boldsymbol{W}\boldsymbol{D_W}^{-1/2})^{-1}\boldsymbol{D_W}^{-1/2}\right)_{ii}\\ &= \left(\boldsymbol{D_W}^{-1/2}\sum_{k=0}^{\infty}(\phi\boldsymbol{D_W}^{-1/2}\boldsymbol{W}\boldsymbol{D_W}^{-1/2})^k\boldsymbol{D_W}^{-1/2}\right)_{ii}.\end{aligned} \tag{E.4.1}$$

Note that the adjacency matrix $\boldsymbol{W}$ is a non-negative matrix, *i.e.*, the element in $\boldsymbol{W}$ is non-negative. Therefore, $\left(\boldsymbol{D_W}^{-1/2}\boldsymbol{W}\boldsymbol{D_W}^{-1/2}\right)^k$ is also non-negative for any $k$. Therefore, we have

$$
\begin{aligned}
&\frac{W_{i+}\Sigma_{ii}}{\sigma_i^2} \\
=&1 + W_{i+}\sum_{k=2}^{\infty}\Big(\boldsymbol{D}_{\boldsymbol{W}}^{-1/2}\big(\phi \boldsymbol{D}_{\boldsymbol{W}}^{-1/2}\boldsymbol{W}\boldsymbol{D}_{\boldsymbol{W}}^{-1/2}\big)^k\boldsymbol{D}_{\boldsymbol{W}}^{-1/2}\Big)_{ii} \\
=&1 + \sum_{k=2}^{\infty}\phi^k \sum_{(v_i,v_{j_1},\ldots,v_{j_l},v_i)\in Path_l^i}\left[\prod_{t=0}^{k}\frac{1}{\sqrt{W_{l_t+}}\sqrt{W_{l_{t+1}+}}}\right]\frac{1}{\sqrt{W_{l_k+}}\sqrt{W_{i+}}}, \qquad \text{(E.4.2)}
\end{aligned}
$$

where $W_{l_0+} = W_{i+}$.

Second, by (5.3.10), we have

$$
\begin{aligned}
\frac{\mathbb{E}_i(T)}{\mathbb{E}_i(\tilde{T})} = \frac{\Sigma_{ii}}{\sigma_i^2/W_{i+}} &= 1 + \sum_{l=2}^{\infty}\phi^l \sum_{(v_i,v_{j_1},\ldots,v_{j_l},v_i)\in Path_l^i}\frac{1}{W_{i+}}\prod_{k=1}^{l}\frac{1}{W_{j_k+}} \\
&\geq 1 + \sum_{l=1}^{\infty}\phi^{2l} \sum_{(v_i,v_{j_1},\ldots,v_{j_l},v_i)\in Path_l^i}\frac{1}{W_{i+}^l}\prod_{k=1}^{l}\frac{1}{W_{j_k+}} \\
&\geq 1 + \sum_{l=1}^{\infty}\phi^{2l} \sum_{((v_i,v_{j_1},\ldots,v_{j_l},v_i)\in Path_l^i}\frac{1}{W_{i+}^l}\frac{1}{W_{i^*+}^l} \\
&= 1 + \sum_{l=1}^{\infty}\phi^{2l}\frac{1}{W_{i^*+}} = \frac{1}{1-\frac{\phi^2}{W_{i^*+}}}.
\end{aligned}
$$

The first inequality holds since we only consider the length 2 paths and their combinations; and the second inequality holds since $W_{j+} \leq W_{i^*+}$ for $j \in N(i)$. We obtain the desired result. □

### E.5 Discussions of Four Nodes Architecture

In this section, we denote $\boldsymbol{W}$ as the transition matrix and $\boldsymbol{P}$ as the change of basis matrix.

### E.5.1 Line Graph

With the line graph, we have the transition matrix $\boldsymbol{W} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 \end{bmatrix}$. Note that the diagonal elements in $\boldsymbol{W}^{2k+1}$ are $0$. Therefore, we only consider the case with $W^{2k}$. We find two relations between the diagonal elements:

$$W_{11}^{2k} = \frac{1}{2}W_{22}^{2k-2}, \quad \text{and} \quad W_{22}^{2k} = \frac{1}{2} + \frac{1}{2}W_{11}^{2k}.$$

Then with $W_{11}^{0} = W_{22}^{0} = 1$, we obtain the solution that $W_{11}^{2k} = \frac{1}{3} + \frac{2}{3}\left(\frac{1}{4}\right)^k$ and $W_{22}^{2k} = \frac{2}{3} + \frac{1}{3}\left(\frac{1}{4}\right)^k$. By symmetry, we have $W_{33}^{2k} = W_{22}^{2k}$ and $W_{44}^{2k} = W_{11}^{2k}$.

Therefore, (1) under the average criterion, we have the improvement:

$$\frac{\sum_{k=0}^{\infty} \phi^{2k}(W_{11}^{2k} + W_{22}^{2k})}{2} = \frac{8 - 5\phi^2}{2(1-\phi^2)(4-\phi^2)} = 1 + \frac{5\phi^2 - 2\phi^4}{2(1-\phi^2)(4-\phi^2)}.$$

(2) under the robust criterion, we have the improvement:

$$\sum_{k=0}^{\infty} \phi^{2k} W_{11}^{2k} = \frac{4 - 3\phi^2}{(1-\phi^2)(4-\phi^2)} = 1 + \frac{2\phi^2 - \phi^4}{(1-\phi^2)(4-\phi^2)}.$$

(3) under the targeted criterion, we have the improvement:

$$\sum_{k=0}^{\infty} \phi^{2k} W_{22}^{2k} = \frac{4 - 2\phi^2}{(1-\phi^2)(4-\phi^2)} = 1 + \frac{3\phi^2 - \phi^4}{(1-\phi^2)(4-\phi^2)}.$$

### E.5.2 Star Graph

With the star graph, we have the transition matrix $\boldsymbol{W} = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$. This is similar to the line graph, whose odd power only contains 0 in diagonal. Thus, we only consider the even power.

$$\boldsymbol{W}^{2k} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}.$$

Therefore, (1) under the average criterion, we have the improvement ratio:

$$\frac{4 + 2\sum_{k=1}^{\infty} \phi^{2k}}{4} = 1 + \frac{\phi^2}{2(1-\phi^2)}.$$

(2) under the robust criterion, we have the improvement:

$$1 + \sum_{k=1}^{\infty} \phi^{2k}/3 = 1 + \frac{\phi^2}{3(1-\phi^2)}.$$

(3) under the targeted criterion, we have the improvement:

$$\sum_{k=0}^{\infty} \phi^{2k} = 1 + \frac{\phi^2}{1-\phi^2}.$$

### E.5.3 Circle Graph

With the circle graph, we have the transition matrix $\boldsymbol{W} = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$. This is similar to the line graph, whose odd power only contains 0 in diagonal. Thus, we only consider the even power.

$$\boldsymbol{W}^{2k} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}.$$

Since circle architecture has symmetric property, all three criteria are the same. We have the following improvement ratio:

$$\frac{4 + 2\sum_{k=1}^{\infty} \phi^{2k}}{4} = 1 + \frac{\phi^2}{2(1-\phi^2)}.$$

### E.5.4 Paw Graph

We have the transition matrix $\boldsymbol{W} = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$. In this case, the problem is more complicated. Note the matrix $\boldsymbol{W}$ can be diagonalized as:

$$\boldsymbol{W} = \boldsymbol{P} \cdot \mathrm{diag}\left(1, \frac{-3-\sqrt{33}}{12}, -\frac{1}{2}, \frac{\sqrt{33}-3}{12}\right) \cdot \boldsymbol{P}^{-1}.$$

Therefore, for $\boldsymbol{W}^k$, we have

$$\boldsymbol{W}^k = \boldsymbol{P} \cdot \text{diag}\left(1, \left(\frac{-3-\sqrt{33}}{12}\right)^k, \left(-\frac{1}{2}\right)^k, \left(\frac{\sqrt{33}-3}{12}\right)^k\right) \cdot \boldsymbol{P}^{-1}.$$

After some matrix algebra manipulations, we obtain the following results: (1) under the average criterion, we have the improvement ratio:

$$\frac{24 - 11\phi^2 - \phi^3}{(1-\phi)(2+\phi)[12 + 6\phi - 2\phi^2]} = 1 + \frac{11\phi^2 + 3\phi^3 - 2\phi^4}{(1-\phi)(2+\phi)[12 + 6\phi - 2\phi^2]}.$$

(2) under the robust criterion, we have the improvement:

$$\frac{6 - 3\phi - 2\phi^2}{(1-\phi)[6 + 3\phi - \phi^2]} = 1 + \frac{2\phi^2 - \phi^3}{(1-\phi)[6 + 3\phi - \phi^2]}.$$

(3) under the targeted criterion, we have the improvement:

$$\frac{6 - 3\phi}{(1-\phi)(6 + 3\phi - \phi^2)} = 1 + \frac{4\phi^2 - \phi^3}{(1-\phi)[6 + 3\phi - \phi^2]}.$$

### E.5.5 Diamond Graph

We have the transition matrix $\boldsymbol{W} = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{bmatrix}$.

Similar to the case with Paw graph, we have

$$\boldsymbol{W} = \boldsymbol{P} \cdot \text{diag}\left(1, -\frac{2}{3}, -\frac{1}{3}, 0\right) \cdot \boldsymbol{P}^{-1}.$$

Therefore, for $\boldsymbol{W}^k$, we have

$$\boldsymbol{W}^k = \boldsymbol{P} \cdot \mathrm{diag}\left(1, \left(-\frac{2}{3}\right)^k, \left(-\frac{1}{3}\right)^k, 0^k\right) \cdot \boldsymbol{P}^{-1}.$$

After matrix algebra manipulation, we obtain the following results: (1) under the average criterion, we have the improvement ratio:

$$1 + \frac{\phi^2(7+3\phi)}{2(9-7\phi^2+2\phi^3)}.$$

(2) under the robust criterion, we have the improvement:

$$\frac{3-\phi-\phi^2}{(1-\phi)(3+2\phi)} = 1 + \frac{\phi^2}{(1-\phi)(3+2\phi)}.$$

(3) under the targeted criterion, we have the improvement:

$$\frac{3(3-\phi^2)}{(1-\phi)(3+\phi)(3+2\phi)} = 1 + \frac{4\phi^2+2\phi^3}{(1-\phi)(3+\phi)(3+2\phi)}.$$

### E.5.6 Fully Connected Graph

In this case, $\boldsymbol{W} = \frac{1}{3}(\mathbf{1}\mathbf{1}^\top - \boldsymbol{I}_4)$. Then we have the diagonal elements in $\boldsymbol{W}^k$ are equal to $\frac{1+3(-1/3)^k}{4}$. Therefore, under all criteria, we have the improvement ratio:

$$\frac{3-2\phi}{(1-\phi)(3+\phi)} = 1 + \frac{\phi^2}{(1-\phi)(3+\phi)}.$$

## REFERENCES


[1] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[2] R. Socher and L. Fei-Fei, "Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, 2010, pp. 966–973.

[3] E. Kidron, Y. Y. Schechner, and M. Elad, "Pixels that sound," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, IEEE, vol. 1, 2005, pp. 88–95.

[4] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 129–136.

[5] R. Arora and K. Livescu, "Kernel CCA for multi-view learning of acoustic features using articulatory measurements.," in *MLSLP*, Citeseer, 2012, pp. 34–37.

[6] S. Bharadwaj, R. Arora, K. Livescu, and M. Hasegawa-Johnson, "Multiview acoustic feature learning using articulatory measurements," in *Intl. Workshop on Stat. Machine Learning for Speech Recognition*, Citeseer, 2012.

[7] A. Vinokourov, J. Shawe-Taylor, and N. Cristianini, "Inferring a semantic representation of text via cross-language correlation analysis," in *Advances in Neural Information Processing Systems*, vol. 1, 2002, pp. 1497–1504.

[8] P. Dhillon, D. P. Foster, and L. H. Ungar, "Multi-view learning of word embeddings via cca," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2011, pp. 199–207.

[9] R. Arora, A. Cotter, K. Livescu, and N. Srebro, "Stochastic optimization for pca and pls," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, IEEE, 2012, pp. 861–868.

[10] H. Abdi, "Partial least square regression (PLS regression)," *Encyclopedia for Research Methods for the Social Sciences*, pp. 792–795, 2003.

[11] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research*, vol. 6, no. 61, pp. 1817–1853, 2005.

[12] R. Arora, P. Mianjy, and T. Marinov, "Stochastic optimization for multiview representation learning using partial least squares," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 1786–1794.

[13] T. Zhao, Z. Wang, and H. Liu, "A nonconvex optimization framework for low rank matrix estimation," in *Advances in Neural Information Processing Systems*, 2015, pp. 559–567.

[14] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via wirtinger flow: Theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.

[15] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points–online stochastic gradient for tensor decomposition," in *Conference on Learning Theory*, 2015, pp. 797–842.

[16] T. T. Cai, X. Li, and Z. Ma, "Optimal rates of convergence for noisy sparse phase retrieval via thresholded wirtinger flow," *The Annals of Statistics*, vol. 44, no. 5, pp. 2221–2251, 2016.

[17] S. N. Ethier and T. G. Kurtz, *Markov Processes: Characterization and Convergence*. John Wiley & Sons, 2009, vol. 282.

[18] C. J. Li, Z. Wang, and H. Liu, "Online ICA: Understanding global dynamics of nonconvex optimization via diffusion processes," in *Advances in Neural Information Processing Systems*, 2016, pp. 4967–4975.

[19] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, no. 6, pp. 459–473, 1989.

[20] Z. Chen, L. F. Yang, C. J. Li, and T. Zhao, "Online partial least square optimization: Dropping convexity for better efficiency and scalability," in *Proceedings of The 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 777–786.

[21] G. H. Golub and C. F. Van Loan, *Matrix Computations*. JHU Press, 2012, vol. 3.

[22] B. Oksendal, *Stochastic Differential Equations: an Introduction with Applications*. Springer Science & Business Media, 2013.

[23] M. B. Cohen, Y. T. Lee, G. Miller, J. Pachocki, and A. Sidford, "Geometric median in nearly linear time," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, ACM, 2016, pp. 9–21.

[24] N. Ross, "Fundamentals of stein's method," *Probab. Surv*, vol. 8, pp. 210–293, 2011.

[25] P. Jain, C. Jin, S. M. Kakade, P. Netrapalli, and A. Sidford, "Streaming PCA: Matching matrix bernstein and near-optimal finite sample guarantees for Oja's algorithm," in *29th Annual Conference on Learning Theory*, 2016, pp. 1147–1164.

[26] O. Shamir, "Fast stochastic algorithms for SVD and PCA: Convergence properties and convexity," in *International Conference on Machine Learning*, 2016, pp. 248–256.

[27] C. J. Li, M. Wang, H. Liu, and T. Zhang, "Near-optimal stochastic approximation for online principal component estimation," *Mathematical Programming*, vol. 167, no. 1, pp. 75–97, 2018.

[28] A. D. Barbour and L. H. Y. Chen, *An Introduction to Stein's Method*. World Scientific, 2005, vol. 4.

[29] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, 10. Springer Series in Statistics New York, 2001, vol. 1.

[30] J. Sun, Q. Qu, and J. Wright, "A geometric analysis of phase retrieval," in *Information Theory (ISIT), 2016 IEEE International Symposium on*, IEEE, 2016, pp. 2379–2383.

[31] S. Bhojanapalli, B. Neyshabur, and N. Srebro, "Global optimality of local search for low rank matrix recovery," in *Advances in Neural Information Processing Systems*, vol. 29, Curran Associates, Inc., 2016, pp. 3880–3888.

[32] X. Li, J. Lu, R. Arora, J. Haupt, H. Liu, Z. Wang, and T. Zhao, "Symmetry, saddle points, and global optimization landscape of nonconvex matrix factorization," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3489–3514, 2019.

[33] R. Ge, J. D. Lee, and T. Ma, "Matrix completion has no spurious local minimum," in *Advances in Neural Information Processing Systems*, 2016, pp. 2973–2981.

[34] Z. Zhu, Q. Li, G. Tang, and M. B. Wakin, "The global optimization geometry of nonsymmetric matrix factorization and sensing," *arXiv preprint arXiv:1703.01256*, 2017.

[35] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*. Springer, 1984, vol. 2.

[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[37] G. Lan, Z. Lu, and R. D. Monteiro, "Primal-dual first-order methods with $\mathcal{O}(1/\epsilon)$ iteration-complexity for cone programming," *Mathematical Programming*, vol. 126, no. 1, pp. 1–29, 2011.

[38] Y. Chen, G. Lan, and Y. Ouyang, "Optimal primal-dual methods for a class of saddle point problems," *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 1779–1814, 2014.

[39] A. Iouditski and Y. Nesterov, "Primal-dual subgradient methods for minimizing uniformly convex functions," *arXiv preprint arXiv:1401.1792*, 2014.

[40] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, IEEE, 1999, pp. 41–48.

[41] R. D. Cook and L. Ni, "Sufficient dimension reduction via inverse regression: A minimum discrepancy approach," *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 410–428, 2005.

[42] G. Gorrell, "Generalized hebbian algorithm for incremental singular value decomposition in natural language processing.," in *11th Conference of the European Chapter of the Association for Computational Linguistics*, Citeseer, vol. 6, 2006, pp. 97–104.

[43] H. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer Science & Business Media, 2003, vol. 35.

[44] R. Ge, C. Jin, P. Netrapalli, and A. Sidford, "Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis," in *International Conference on Machine Learning*, 2016, pp. 2741–2750.

[45] Z. Allen-Zhu and Y. Li, "Doubly accelerated methods for faster CCA and generalized eigendecomposition," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, International Convention Centre, Sydney, Australia: PMLR, 2017, pp. 98–106.

[46] R. Arora, T. V. Marinov, P. Mianjy, and N. Srebro, "Stochastic approximation for canonical correlation analysis," in *Advances in Neural Information Processing Systems*, 2017, pp. 4778–4787.

[47] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Procesing Systems*, 2000, pp. 1057–1063.

[48] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2009.

[49] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Procesing Systems*, 2014, pp. 2672–2680.

[50] B. T. Polyak, "Gradient methods for minimizing functionals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 3, no. 4, pp. 643–653, 1963.

[51] Z.-Q. Luo and P. Tseng, "Error bounds and convergence analysis of feasible descent methods: A general approach," *Annals of Operations Research*, vol. 46, no. 1, pp. 157–178, 1993.

[52] D. S. Dummit and R. M. Foote, *Abstract Algebra*. Wiley Hoboken, 2004, vol. 3.

[53] J. L. Doob, "The brownian movement and stochastic equations," *Annals of Mathematics*, pp. 351–369, 1942.

[54] B. Mishra and R. Sepulchre, "Riemannian preconditioning," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 635–660, 2016.

[55] S. Mak, C.-L. Sung, X. Wang, S.-T. Yeh, Y.-H. Chang, V. R. Joseph, V. Yang, and C.-F. J. Wu, "An efficient surrogate model for emulation and physics extraction of large eddy simulations," *Journal of the American Statistical Association*, vol. 113, no. 524, pp. 1443–1456, 2018.

[56] T. Dasgupta, C. Ma, V. R. Joseph, Z. Wang, and C. F. J. Wu, "Statistical modeling and analysis for robust synthesis of nanostructures," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 594–603, 2008.

[57] J. Chen, S. Mak, V. R. Joseph, and C. Zhang, "Function-on-function kriging, with applications to three-dimensional printing of aortic tissues," *Technometrics*, pp. 1–12, 2020.

[58] M. Kearns and S. Singh, "Near-optimal reinforcement learning in polynomial time," *Machine Learning*, vol. 49, no. 2-3, pp. 209–232, 2002.

[59] T. J. Santner, B. J. Williams, W. Notz, and B. J. Williams, *The Design and Analysis of Computer Experiments*. (2nd. edn.) Springer, 2018.

[60] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of Bayesian methods for seeking the extremum," *Towards Global Optimization*, vol. 2, no. 2, pp. 117–129, 1978.

[61] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[62] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10, Haifa, Israel: Omnipress, 2010, pp. 1015–1022, ISBN: 978-1-60558-907-7.

[63] P. I. Frazier, W. B. Powell, and S. Dayanik, "A knowledge-gradient policy for sequential information collection," *SIAM Journal on Control and Optimization*, vol. 47, no. 5, pp. 2410–2439, 2008.

[64] W. Scott, P. Frazier, and W. Powell, "The correlated knowledge gradient for simulation optimization of continuous parameters using Gaussian process regression," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 996–1026, 2011.

[65] Y. Zhang, Z.-H. Han, and K.-S. Zhang, "Variable-fidelity expected improvement method for efficient global optimization of expensive functions," *Structural and Multidisciplinary Optimization*, vol. 58, no. 4, pp. 1431–1451, 2018.

[66] P. Feliot, J. Bect, and E. Vazquez, "A Bayesian approach to constrained single-and multi-objective optimization," *Journal of Global Optimization*, vol. 67, no. 1-2, pp. 97–133, 2017.

[67] S. Marmin, C. Chevalier, and D. Ginsbourger, "Differentiating the multipoint expected improvement for optimal batch design," in *International Workshop on Machine Learning, Optimization and Big Data*, Springer, 2015, pp. 37–48.

[68] C. Qin, D. Klabjan, and D. Russo, "Improving the expected improvement algorithm," in *Advances in Neural Information Processing Systems*, 2017, pp. 5381–5391.

[69] A. D. Bull, "Convergence rates of efficient global optimization algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2879–2904, 2011.

[70] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, 2012, pp. 2951–2959.

[71] R.-B. Chen, W. Wang, and C.-F. J. Wu, "Sequential designs based on Bayesian uncertainty quantification in sparse representation surrogate modeling," *Technometrics*, vol. 59, no. 2, pp. 139–152, 2017.

[72] M. S. Handcock and M. L. Stein, "A Bayesian analysis of kriging," *Technometrics*, vol. 35, no. 4, pp. 403–410, 1993.

[73] R. Benassi, J. Bect, and E. Vazquez, "Robust Gaussian process-based global optimization using a fully Bayesian expected improvement criterion," in *International Conference on Learning and Intelligent Optimization*, Springer, 2011, pp. 176–190.

[74] H. Wackernagel, *Multivariate Geostatistics: an Introduction with Applications*. Springer Science & Business Media, 1995.

[75] R. A. Olea, *Geostatistics for Engineers and Earth Scientists*. Springer Science & Business Media, 2012.

[76] A. Gelman, "Prior distributions for variance parameters in hierarchical models," *Bayesian Analysis*, vol. 1, no. 3, pp. 515–534, 2006.

[77] B. P. Carlin and T. A. Louis, *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall/CRC Boca Raton, 2000, vol. 88.

[78] A. Doucet, S. J. Godsill, and C. P. Robert, "Marginal maximum a posteriori estimation using Markov chain Monte Carlo," *Statistics and Computing*, vol. 12, no. 1, pp. 77–84, 2002.

[79] Q. Liu and A. Ihler, "Variational algorithms for marginal MAP," *Journal of Machine Learning Research*, vol. 14, pp. 3165–3200, 2013.

[80] D. Xiu, *Numerical Methods for Stochastic Computations: a Spectral Method Approach*. Princeton University Press, 2010.

[81] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[82] M. D. Morris and T. J. Mitchell, "Exploratory designs for computational experiments," *Journal of Statistical Planning and Inference*, vol. 43, no. 3, pp. 381–402, 1995.

[83] R. Lekivetz and B. Jones, “Fast flexible space-filling designs for nonrectangular regions,” *Quality and Reliability Engineering International*, vol. 31, no. 5, pp. 829–837, 2015.

[84] S. Mak and V. R. Joseph, “Minimax and minimax projection designs using clustering,” *Journal of Computational and Graphical Statistics*, vol. 27, no. 1, pp. 166–178, 2018.

[85] V. R. Joseph, E. Gul, and S. Ba, “Designing computer experiments with multiple types of factors: The MaxPro approach,” *Journal of Quality Technology*, pp. 1–12, 2019.

[86] J. L. Loeppky, J. Sacks, and W. J. Welch, “Choosing the sample size of a computer experiment: A practical guide,” *Technometrics*, vol. 51, no. 4, pp. 366–376, 2009.

[87] H. Wendland, *Scattered Data Approximation. Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, 2004, vol. 17.

[88] N. Cressie, *Statistics for Spatial Data*. New York: Wiley, 1991.

[89] I. Castillo, J. Schmidt-Hieber, and A. Van der Vaart, “Bayesian linear regression with sparse priors,” *The Annals of Statistics*, vol. 43, no. 5, pp. 1986–2018, 2015.

[90] G Wynne, F. Briol, and M Girolami, “Convergence guarantees for Gaussian process means with misspecified likelihoods and smoothness,” *arXiv preprint arXiv:2001.10818*, 2020.

[91] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, 1992.

[92] S. Surjanovic and D. Bingham, *Virtual library of simulation experiments: Test functions and datasets*, Retrieved August 16, 2019, from `http://www.sfu.ca/~ssurjano`, 2015.

[93] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[94] Y. Deville, D. Ginsbourger, and O. Roustant, *Kergp: Gaussian process laboratory*, `https://cran.r-project.org/web/packages/kergp`, 2019.

[95] S. Mak and C.-F. J. Wu, “Analysis-of-marginal-tail-means (ATM): A robust method for discrete black-box optimization,” *Technometrics*, vol. 61, no. 4, pp. 545–559, 2019.

[96] R. Jin, C.-J. Chang, and J. Shi, “Sequential measurement strategy for wafer geometric profile estimation,” *IIE Transactions*, vol. 44, no. 1, pp. 1–12, 2012.

[97] R Singh, M Fakhruddin, and K. Poole, “Rapid photothermal processing as a semiconductor manufacturing technology for the 21st century,” *Applied Surface Science*, vol. 168, no. 1-4, pp. 198–203, 2000.

[98] T. A. Brunner, V. C. Menon, C. W. Wong, O. Gluschenkov, M. P. Belyansky, N. M. Felix, C. P. Ausschnitt, P. Vukkadala, S. Veeraraghavan, and J. K. Sinha, “Characterization of wafer geometry and overlay error on silicon wafers with nonuniform stress,” *Journal of Micro/Nanolithography, MEMS, and MOEMS*, vol. 12, no. 4, pp. 1–13, 2013.

[99] COMSOL, *COMSOL Multiphysics© v. 5.4.* 2018.

[100] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[101] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[102] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[103] W. Liu, Y.-M. Zhang, X. Li, Z. Yu, B. Dai, T. Zhao, and L. Song, “Deep hyperspherical learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 3950–3960.

[104] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *arXiv preprint arXiv:1312.6199*, 2013.

[105] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint arXiv:1412.6572*, 2014.

[106] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436.

[107] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4480–4488.

[108] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[109] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2017, pp. 39–57.

[110] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[111] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[112] R. Gao and A. J. Kleywegt, "Distributionally robust stochastic optimization with wasserstein distance," *arXiv preprint arXiv:1604.02199*, 2016.

[113] H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review," *arXiv preprint arXiv:1908.05659*, 2019.

[114] W. X. Haichao Zhang, *Adversarial interpolation training: A simple approach for improving model robustness*, 2019.

[115] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[116] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of Operations Research*, vol. 153, no. 1, pp. 235–256, 2007.

[117] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong, "Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks," *arXiv preprint arXiv:1905.00441*, 2019.

[118] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17, Sydney, NSW, Australia: JMLR.org, 2017, pp. 1126–1135.

[119] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[120] J. Schmidhuber, “Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-... hook,” PhD thesis, Technische Universität München, 1987.

[121] ——, “Learning to control fast-weight memories: An alternative to dynamic recurrent networks,” *Neural Computation*, vol. 4, no. 1, pp. 131–139, 1992.

[122] ——, “A neural network that embeds its own meta-levels,” in *Neural Networks, 1993., IEEE International Conference on*, IEEE, 1993, pp. 407–412.

[123] A. S. Younger, S. Hochreiter, and P. R. Conwell, “Meta-learning with backpropagation,” in *Neural Networks, 2001. Proceedings. IJCNN’01. International Joint Conference on*, IEEE, vol. 3, 2001, pp. 2001–2006.

[124] S. Hochreiter, A. S. Younger, and P. R. Conwell, “Learning to learn using gradient descent,” in *International Conference on Artificial Neural Networks*, Springer, 2001, pp. 87–94.

[125] M. Andrychowicz, M. Denil, S. G. Colmenarejo, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. de Freitas, “Learning to learn by gradient descent by gradient descent,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Curran Associates Inc., 2016, pp. 3988–3996.

[126] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, “Boosting adversarial attacks with momentum,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9185–9193.

[127] P. Tabacof and E. Valle, “Exploring the space of adversarial images,” in *2016 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2016, pp. 426–433.

[128] S. Zagoruyko and N. Komodakis, “Wide residual networks,” *arXiv preprint arXiv:1605.07146*, 2016.

[129] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, “Theoretically principled trade-off between robustness and accuracy,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 7472–7482.

[130] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *International Conference on Learning Representations*, 2018.

[131] Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into transferable adversarial examples and black-box attacks,” *International Conference on Learning Representations*, 2017.

[132] T. Liu, Z. Chen, E. Zhou, and T. Zhao, “Toward deeper understanding of non-convex stochastic optimization with momentum using diffusion approximations,” *arXiv preprint arXiv:1802.05155*, 2018.

[133] A. Krogh and J. A. Hertz, “A simple weight decay can improve generalization,” in *Advances in Neural Information Processing Systems*, 1992, pp. 950–957.

[134] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[135] B. Dai, N. He, Y. Pan, B. Boots, and L. Song, “Learning from conditional distributions via dual embeddings,” in *Artificial Intelligence and Statistics*, PMLR, 2017, pp. 1458–1467.

[136] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update ReLU converge to a local nash equilibrium,” in *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637.

[137] S. Baluja and I. Fischer, “Adversarial transformation networks: Learning to generate adversarial examples,” *arXiv preprint arXiv:1703.09387*, 2017.

[138] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, “Generating adversarial examples with adversarial networks,” *arXiv preprint arXiv:1801.02610*, 2018.

[139] A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 219–230, 2004.

[140] M. Valero, J. Clemente, G. Kamath, Y. Xie, F.-C. Lin, and W. Song, “Real-time ambient noise subsurface imaging in distributed sensor networks,” in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, IEEE, 2017, pp. 1–8.

[141] L. Peel and A. Clauset, “Detecting change points in the large-scale structure of evolving networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015.

[142] Y. Wang, A. Chakrabarti, D. Sivakoff, and S. Parthasarathy, “Fast change point detection on dynamic social networks,” *arXiv preprint arXiv:1705.07325*, 2017.

[143] B. B. Averbeck, P. E. Latham, and A. Pouget, “Neural correlations, population coding and computation,” *Nature Reviews Neuroscience*, vol. 7, no. 5, pp. 358–366, 2006.

[144] A. Ghafouri, Y. Vorobeychik, and X. Koutsoukos, "Adversarial regression for detecting attacks in cyber-physical systems," *arXiv preprint arXiv:1804.11022*, 2018.

[145] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[146] A. N. Shiryaev, "On optimum methods in quickest detection problems," *Theory of Probability & Its Applications*, vol. 8, no. 1, pp. 22–46, 1963.

[147] S. Roberts, "A comparison of some control chart procedures," *Technometrics*, vol. 8, no. 3, pp. 411–430, 1966.

[148] A. G. Tartakovsky and V. V. Veeravalli, "An efficient sequential procedure for detecting changes in multichannel and distributed systems," in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997)*, IEEE, vol. 1, 2002, pp. 41–48.

[149] M. Basseville, "A discussion on 'detection of intrusions in information systems by sequential change-point methods' by Tartakovsky, Rozovskii, Blažek, and Kim," *Statistical Methodology*, vol. 3, no. 3, pp. 299 –303, 2006.

[150] Y. Mei, "Efficient scalable schemes for monitoring a large number of data streams," *Biometrika*, vol. 97, no. 2, pp. 419–433, 2010.

[151] W. Jiang, S. W. Han, K.-L. Tsui, and W. H. Woodall, "Spatiotemporal surveillance methods in the presence of spatial correlation," *Statistics in Medicine*, vol. 30, no. 5, pp. 569–583, 2011.

[152] Y. Xie and D. Siegmund, "Spectrum opportunity detection with weak and correlated signals," in *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, IEEE, 2012, pp. 128–132.

[153] M. L. Lee, D. Goldsman, S.-H. Kim, and K.-L. Tsui, "Spatiotemporal biosurveillance with spatial clusters: Control limit approximation and impact of spatial correlation," *IIE Transactions*, vol. 46, no. 8, pp. 813–827, 2014.

[154] M. L. Lee, D. Goldsman, and S.-H. Kim, "Robust distribution-free multivariate cusum charts for spatiotemporal biosurveillance in the presence of spatial correlation," *IIE Transactions on Healthcare Systems Engineering*, vol. 5, no. 2, pp. 74–88, 2015.

[155] J. Chen, S.-H. Kim, and Y. Xie, "$S^3T$: An efficient score-statistic for spatio-temporal surveillance," *arXiv preprint arXiv:1706.05331*, 2017.

[156] A. G. Tartakovsky and V. V. Veeravalli, “Asymptotically optimal quickest change detection in distributed sensor systems,” *Sequential Analysis*, vol. 27, no. 4, pp. 441–475, 2008.

[157] O. Hadjiliadis, H. Zhang, and H. V. Poor, “One shot schemes for decentralized quickest change detection,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3346–3359, 2009.

[158] Q. Liu, R. Zhang, and Y. Xie, “Distributed change detection via average consensus over networks,” in *Workshop on Stochastic Models, Statistics and their Application*, Springer, 2019, pp. 177–192.

[159] J. Besag, “Spatial interaction and the statistical analysis of lattice systems,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, pp. 192–225, 1974.

[160] D Brook, “On the distinction between the conditional probability and the joint probability approaches in the specification of nearest-neighbour systems,” *Biometrika*, vol. 51, no. 3/4, pp. 481–483, 1964.

[161] Y. Xie and D. Siegmund, “Sequential multi-sensor change-point detection,” in *2013 Information Theory and Applications Workshop (ITA)*, IEEE, 2013, pp. 1–20.

[162] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*. Springer Science & Business Media, 2006.

[163] F. Li, R. Xie, B. Yang, L. Guo, P. Ma, J. Shi, J. Ye, and W. Song, “Detection and identification of cyber and physical attacks on distribution power grids with pvs: An online high-dimensional data-driven approach,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 2019.

[164] V. V. Veeravalli and T. Banerjee, “Quickest change detection,” in *Academic Press Library in Signal Processing*, vol. 3, Elsevier, 2014, pp. 209–255.

[165] H. M. Merrill and F. C. Schweppe, “Bad data suppression in power system static state estimation,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-90, no. 6, pp. 2718–2725, 1971.

[166] A. B. Lawson, A. Biggeri, D. Böhning, E. Lesaffre, J.-F. Viel, A. Clark, P. Schlattmann, and F. Divino, “Disease mapping models: An empirical evaluation. disease mapping collaborative group,” *Statistics in Medicine*, vol. 19, no. 17, pp. 2217–2241, 2000.

[167] P. Elliott and D. Wartenberg, "Spatial epidemiology: Current approaches and future challenges," *Environmental Health Perspectives*, vol. 112, no. 9, pp. 998–1006, 2004.

[168] S. Banerjee, B. P. Carlin, and A. E. Gelfand, *Hierarchical Modeling and Analysis for Spatial Data*. Chapman and Hall/CRC, 2014.

[169] O. Parent and J. P. LeSage, "Using the variance structure of the conditional autoregressive spatial specification to model knowledge spillovers," *Journal of Applied Econometrics*, vol. 23, no. 2, pp. 235–256, 2008.

[170] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT press, 2009.

[171] S. I. Resnick, *Adventures in Stochastic Processes*. Springer Science & Business Media, 1992.

[172] N. Cressie and C. K. Wikle, *Statistics for Spatio-temporal Data*. John Wiley & Sons, 2015.

[173] R. Faudree, E. Flandrin, and Z. Ryjáček, "Claw-free graphs—a survey," *Discrete Mathematics*, vol. 164, no. 1-3, pp. 87–147, 1997.

[174] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, no. 1, pp. 17–60, 1960.

[175] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 1, pp. 440–442, 1998.

[176] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*. CRC press, 2004.

[177] P. Foldiak, "Sparse coding in the primate cortex," *The Handbook of Brain Theory and Neural Networks*, 2003.

[178] A. P. Davison, D. Brüderle, J. M. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, "Pynn: A common interface for neuronal network simulators," *Frontiers in Neuroinformatics*, vol. 2, p. 11, 2009.

[179] M. L. Hines and N. T. Carnevale, "Neuron: A tool for neuroscientists," *The Neuroscientist*, vol. 7, no. 2, pp. 123–135, 2001.

[180] V. Dakos, S. R. Carpenter, W. A. Brock, A. M. Ellison, V. Guttal, A. R. Ives, S. Kéfi, V. Livina, D. A. Seekell, E. H. van Nes, *et al.*, "Methods for detecting early

warnings of critical transitions in time series illustrated using simulated ecological data," *PloS one*, vol. 7, no. 7, e41010, 2012.

[181] L. C. Evans, "Partial differential equations," *Graduate studies in mathematics*, vol. 19, no. 2, 1998.

[182] B. D. Nowakowski, "On multi-parameter semimartingales, their integrals and weak convergence," *Doctoral dissertation, The Pennsylvania State University*, 2013.

[183] T. Wenzel, G. Santin, and B. Haasdonk, "A novel class of stabilized greedy kernel approximation algorithms: Convergence, stability and uniform point distribution," *Journal of Approximation Theory*, p. 105 508, 2020.

[184] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, PMLR, 2018, pp. 274–283.

[185] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," *arXiv preprint arXiv:1902.06705*, 2019.

[186] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-gan: Protecting classifiers against adversarial attacks using generative models," *arXiv preprint arXiv:1805.066 05*, 2018.

[187] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16, Barcelona, Spain: Curran Associates Inc., 2016, pp. 4572–4580, ISBN: 9781510838819.

[188] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[189] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 1889–1897.

[190] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[191] A. G. Tartakovsky and V. V. Veeravalli, "Change-point detection in multichannel and distributed systems," *Applied Sequential Methodologies: Real-World Examples with Data Analysis*, vol. 173, pp. 339–370, 2004.

[192] H. V. Poor and O. Hadjiliadis, *Quickest Detection*. Cambridge University Press, 2008.

[193] D. Siegmund, *Sequential Analysis: Tests and Confidence Intervals*. Springer Science & Business Media, 1985.

[194] E. M. Stein and R. Shakarchi, *Princeton Lectures in Analysis*. Princeton University Press, 2003.